2

FINAL REPORT

Office of Naval Research
Optical Computing Strategies
Contract N00014-86-K-0591

2

FINAL REPORT

Office of Naval Research
Optical Computing Strategies
Contract N00014-86-K-0591

Submitted by

H. J. Caulfield
Principal Investigator
Center for Applied Optics
University of Alabama in Huntsville
Huntsville, Alabama 35899

February, 1990

CONTENTS

"Solving Eigenvalue Problems Using the Bimodal Optical Computer," M. A. G. Abushagur and H. J. Caulfield, unpublished.

"Solving System of Linear Equations Using the Bimodal Optical Computer (Experimental Results)," M. A. Habli, M.A.G. Abushagur, H. J. Caulfield, SPIE 936, (1988) pp 315.

APPENDIX B
MASSIVE HOLOGRAPHIC INTERCONNECTION - Selected Papers

I. OVERVIEW

II. SELECTED PAPERS

"Parallel N^4 Weighted Optical Interconnections," H. J. Caulfield, Appl. Opt. 26, No. 19 (1987) pp 4039.

"A Breakthrough for Optical Neural Nets," Lasers & Optronics, Lasers and Optronics pp 22.

"Massive Holographic Interconnection Networks and Their Limitations," J. Shamir, H. J. Caulfield, and R. B. Johnson, Appl. Opt. 28, No. 2, (1989) pp 311.

APPENDIX C
APPLICATIONS PAPERS

I. OVERVIEW

II. SELECTED PAPERS

"Systolic Optical Cellular Array Processors," H. J. Caulfield, Opt. Eng. 25, (1986) pp 825.

"Optical Interconnection Based Symbolic Manipulations," H. J. Caulfield, Opt. Eng. 25, (1986) pp 1179.

"Massively Parallel Optical Data Base Management," H. J. Caulfield, SPIE 938, (1988) pp 52.

"Optical Database/Knowledgebase Machines," P. Bruce Berra, K-Heinz Brenner, W. T. Cathey, H. J. Caulfield, S. H. Lee, and Harold Szu, Appl. Opt. 29, (1990) pp 195.

"Page Oriented Holographic Memories and Optical Pattern Recognition," H. J. Caulfield, SPIE 754, (1987) pp 74.

"Page Oriented Holographic Memory Addressing of Optical Bistable Devices Arrays," H. J. Caulfield, SPIE 769 (1986) pp 101.

"Optical Mapping Applications," H. J. Caulfield, SPIE 881, (1988) pp 56.

"Stacked Page Oriented Holographic Memory," H. J. Caulfield, SPIE 883 (1988) pp 203.

APPENDIX D
MISCELLANEOUS APPLICATIONS

I. OVERVIEW

II. SELECTED PAPERS

"Optical Computing and the Fredkin Gates," J. Shamir, H. J. Caulfield, W. Micelli and R. J. Seymour, Appl. Opt. 25, (1986) pp 1604.

"Optical Fredkin Gate," J. Shamir and H. J. Caulfield, SPIE 625 (1986), pp 2.

"High-efficiency Rapidly Programmable Optical Interconnections," J. Shamir and H. J. Caulfield, Appl. Opt. 26, (1987) pp 1032.

"Three-dimensional Optical Interconnection Gate Array," Joseph Shamir, Appl. Opt. 26 (1987) pp 3455.

"Residue Arithmetic Processing Utilizing Optical Fredkin Gate Arrays," M. M. Mirsalehi, J. Shamir, and H. J. Caulfield, Appl. Opt. 26, (1987) pp 3940.

"Pattern Recognition Using Reduced Information Content Filters," J. Shamir, H. J. Caulfield, and J. Rosen, Appl. Opt. 26, (1987) pp 2311.

"Rotation-Invarient Pattern Recognition and Some of Its Limitations," H. J. Caulfield and J. Shamir, SPIE 613, (1986) pp 260.

"Circular Harmonic Phase Filters for Efficient Rotation-Invariant Pattern Recognition," Appl. Opt. 27, (1988) pp 2895.

"Weight Discretization in Backward Error Propagation Neural Networks," E. Fiesler, A. Choudry, and H. J. Caulfield, accepted for publication by IEEE Transactions on Systems, Man, and Cybernetics.

"Some Theoretical Upperbounds on the Capacity of Neural Networks," E. Fiesler, H. J. Caulfield, and A. Choudry, Proc. WNN-AIND 90, Vol. II (1990).

"Maximum Capacity Topologies for Fully Connected Layered Neural Networks with Bi-Directional Connections," E. Fiesler, H. J. Caulfield, A. Choudry, and J. P. Ryan, submitted to IJCNN, July, 1990.

"Fundamental Speed Limitations on Parallel Processing," Joseph Shamir, Rapid Communications, Appl. Opt. 26, (1987) pp 1567.

STATEMENT "A" per Michael Karp
ONRRR-Atlanta
TELECON 3/9/90 VG

APPENDIX E
OPTICAL SYSTOLIC ARRAY PROCESSING USING A NOVEL INTEGRATED ACOUSTO-OPTIC MODULATOR MODULE

(TSAI ASSOCIATES)

APPENDIX F
POLYNOMIAL CONVOLUTION ALGORITHM FOR MATRIX MULTIPLICATION WITH APPLICATION FOR OPTICAL COMPUTING

(HARVARD UNIVERSITY)

I. INTRODUCTION

This report covers a very diverse, multi-year effort to explore and develop the role of optical computing for SDI purposes. Part of this effort was through subcontractors whose final reports are separately appended. Other parts of this work, were involved in efforts to unify and publicize the activity of SDI in optical computing. We believe this effort was important in counteracting the assertions made by disgruntled scientists in other fields that SDI funding was only for "mediocre scientists." The ~~UAH~~ effort was primarily in two fields: Optical Algebra and Massive Parallel Holographic Interconnection. In addition to that, there was work on a variety *of other activities such as pattern* recognition, optical interconnection, and low energy optical computing. This report will attempt to organize, capsulize, and comment upon those various activities. In addition, we include some of the relevant technical documents as appendixes in order to provide more detail for those who wish to have it. Finally, we offer a program wrap-up which demonstrates quite conclusively that the effort under this contract was not only fruitful, but also generative of considerable current and future activity. Thus, this program has planted seeds that will lead, in a significant degree, to the accomplishment of the original goal of making optical computing useful for SDI and for America.

II. OPTICAL ALGEBRA

Optics has been suggested for algebra for many years, because there are geometry in configurations which permit it to be done extremely rapidly. Before our work, the major problem with optical algebra has been ~~was~~ that high accuracy was essentially unobtainable. One of the primary goals of our program was to show that one ~~we~~ could use low accuracy optics for the computationally intense part of algebraic computations and bootstrap the accuracy with moderately high accuracy digital electronics in very simple, hard-wired configurations. [illegible] Every technical library has many shelves full of books on numerical algebra. All of them assure the reader that low accuracy processors are worthless in obtaining even moderately accurate results for any realistic problem and that for ill-conditions or singular equation sets, low accuracy processors are worthless. If we believe the results of the great mathematicians, who wrote those books, it is clear that optical algebra is doomed unless it is possible to somehow change the rules or change the problems. There follows an account of exactly how we did that.

According to various estimates, somewhere between 50% and 75% of all CPU time in the United States is spent in solving some sort of linear algebra. Examples include least squares analysis, antenna beam steering, linear regression, computational fluid dynamics, finite element analysis, or simply N linear equations with N unknowns.

Other <u>nonlinear</u> <u>algebra</u> problems are also important. These include image processing, linear programming, and super resolution.

To the extent that optics can solve such problems in a parallel fashion, it can lead to small fast processors which would greatly improve the utility of trackers, radar, sonar, etc.

<u>WHAT IS THE CURRENT STATUS?</u>

We want to solve problems like

$$2x_1 + 3x_2 + x_3 = 4$$

$$3x_1 + x_2 + 3x_3 = 2$$

$$3x_1 + 4x_2 + 7x_3 = 1$$

We can represent these generally as

$$\mathbf{A}\,x = \mathbf{b}.$$

In this case

$$A = \begin{bmatrix} 2 & 3 & 1 \\ 3 & 1 & 1 \\ 3 & 4 & 7 \end{bmatrix}$$

$$\mathbf{b} = \begin{bmatrix} 4 \\ 2 \\ 1 \end{bmatrix} \quad \text{and} \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

The matrix A and the vector **b** are given. We seek the vector **x**.

There is a way to assign a single number (a "norm") to vectors and matrices. We normally use the Euclidean norm, e.g.

$$\| \mathbf{x} \| = \left[x_1^2 + x_2^2 + x_3^2 \right]^{\frac{1}{2}}$$

The word "solve" has two different meanings. We presume there is a "true" answer x_T. We can say we have an ϵ - accurate solution if

$$\| \mathbf{x} - \mathbf{x}_T \| < \epsilon .$$

A weaker sense of "solve" is

$$\| \mathbf{b} - A\, \mathbf{x} \| < \epsilon .$$

This is weaker in the rough sense that some good solutions in this sense may not be close to $\mathbf{x}_T$. On the other hand, for many problems, this "low residual" solution is perfectly adequate. The Bimodal Optical Computer (BOC) minimizes the residual.

One speaks of computational complexity in terms of how something scales with some resource. We will speak of spatial and temporal complexity. We will represent an N x N matrix in parallel using N^2 numbers. We say the spatial complexity scales on the order of N^2, written $O(N^2)$. We will show that the temporal complexity is O(1), i.e., independent of N, provided that N is small enough to be represented spatially in our processor.

The most basic concepts are over a century old (due to Lord Kelvin).

(1) We use a fast, low-accuracy processor to obtain a first guess $\mathbf{x}_0$,

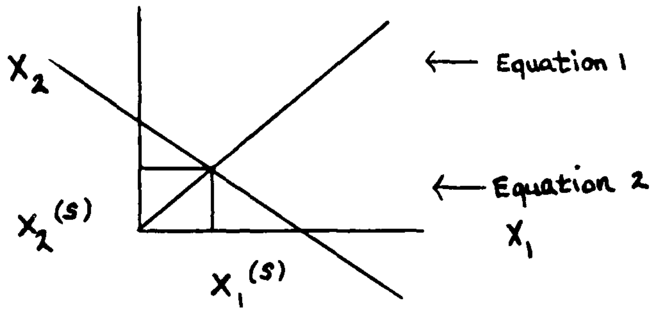
(2) We use a slow, accurate processor to evaluate the residual $\mathbf{r}_0 = \mathbf{b} - A\ \mathbf{x}_0$, if $\| \mathbf{r}_0 \| < \epsilon$, stop.

(3) Otherwise, use the low accuracy solver to solve for $\Delta\, \mathbf{x}_0 = \mathbf{r}_0$. If we could solve that problem curately, then $\mathbf{x}, = \mathbf{x}_0 + \Delta\, \mathbf{x}_0$ would have zero residual. Thus,

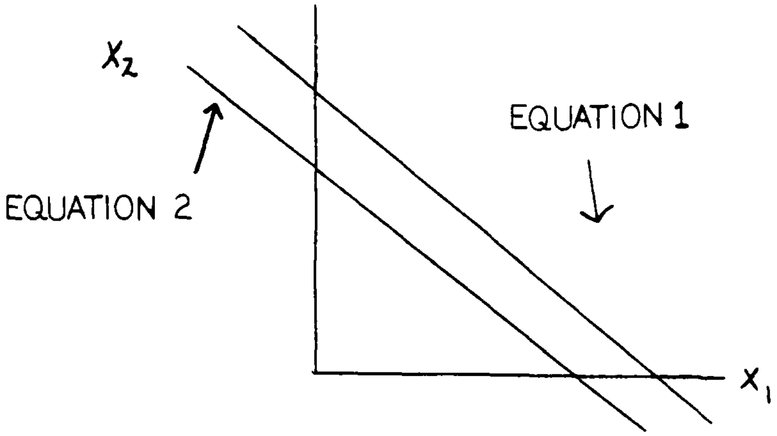
$$\begin{aligned} A\ \mathbf{x}_1 &= A\ (\mathbf{x}_0 + \Delta\ \mathbf{x}_0) \\ &= A\ \mathbf{x}_0 + A\ \Delta\ \mathbf{x}_0 \\ &= A\ \mathbf{x}_0 + \mathbf{r}_0 \\ &= A\ \mathbf{x}_0 + \mathbf{b} - A\ \mathbf{x}_0 \\ &= \mathbf{b}. \end{aligned}$$

(4) Use the slow, accurate processor to evaluate $\mathbf{r}_1 = \mathbf{b} - A\ \mathbf{x}_1$, if $\| \mathbf{r}_1 \| < \epsilon$, stop. Otherwise go to (3).

Some algebra problems resist accurate solution more than others. In high school, we solved N=2 problems graphically.



The solution is $x_1^{(s)}$, $x_2^{(s)}$. Problems like this are said to be "well conditioned" and are quite rare in real life. A more common case is



Such problems are said to be "ill conditioned." If the lines are parallel, we say A is "singular." Let us now make this somewhat more rigorous. Let us define a "condition number"

$$\chi (A) = \| A \| \cdot \| A^{-1} \| .$$

Then

$$\epsilon (\| \mathbf{x} \|) = \chi (A) \ \epsilon (P),$$

where

$\epsilon (\| \mathbf{x} \|)$ = relative error in the result and

$\epsilon (P)$ = relative accuracy of the processor.

If we have ϵ (P) = 0.1 (very good optics) and χ (A) = 10 (wonderfully benign problem),

$$\epsilon \; (\| \mathbf{x} \|) = 1,$$

i.e., 100% errors are likely.

This is why we go to 32 bit floating point electronics. No one wants an answer accurate to one part in 2^{32} (~ 4 x 10^{9}). We need that to get meaningful answers for large χ. The ultimate ill-conditioning, singularity, corresponds to infinite χ. Such problems are common.

In roughly 1985, Caulfield showed that this iterative process converges (roughly) if

$$\epsilon \; (P) < \frac{1}{2\chi(A)} \quad ,$$

For good optics, ϵ (P) = 0.1. Thus we need

$$\chi \; (A) < 5$$

to guarantee solution. This is silly. No real problems are so benign.

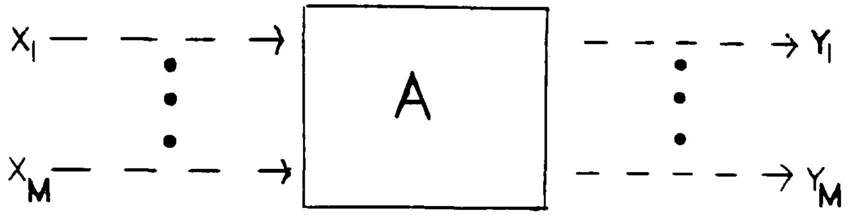
In 1987, we showed that replacing A by A' = A + E where E is an error matrix and

$$\| E \| / \| A \| \| \ll 1,$$

leads to convergence for all problems independently of χ. For large χ, the x which minimizes $\| \mathbf{r} \|$ may be less close to $\mathbf{x}_T$ than would be the case for small χ. Nevertheless, we can drive $\| \mathbf{r} \|$ to zero in very few iterations even for singular matrices. Call this Breakthrough 1.

To do the fast, low-accuracy solution O (1) in time; we

To do the fast, low-accuracy solution O (1) in time; we use another trick. We employ a parallel A **x** = **y** device.



These are easy in optics. Wai Cheng and Caulfield showed that if we correct x_k, with a signal proportional to $b_k - y_k$, for all k, then this system would "relax from any starting **x** to one satisfying A **x** = **b** (in the low $\| \mathbf{r} \|$ sense) under the circumstance that A is "positive definite." To explain this, we need one more diversion.

A vector e such that

$$A \mathbf{e} = \lambda \mathbf{e},$$

where λ is a scalar, is said to be an "eigenvalue." Let us arrange the eigenvalues of A such that

$$\lambda, < \lambda_2 < \cdot \cdot \cdot < \lambda_r$$

(r connotes "rank," a concept we choose not to define here). Interestingly,

$$\chi \text{ (A)} = 1 \lambda_r |/|\lambda_1| .$$

The interesting thing for our purposes is that the relaxation processor converges at a rate (roughly) of

$$e^{-\lambda_1 t}.$$

Obviously if λ, > 0, it does not converge. Here t is normalized by the round trip time in the system. A matrix for which λ, > 0, is said to be positive definite. A matrix

$$B = \begin{bmatrix} 12 \\ 34 \end{bmatrix}$$

can undergo a row-for-column switch to form a transpose

$$B^T = \begin{bmatrix} 13 \\ 24 \end{bmatrix}.$$

Since the matrix elements may be complex, we can complex conjugate a matrix A to get A*. We call

$$(A^*)^T = (A^T)^* = A^H,$$

the Hermitian of A. For any matrix both AA^H and A^HA are nonnegative definite ($\lambda_1 > 0$). We noted that $A^HA + E$ and $AA^H + E$ are positive definite if $E > 0$.

Note, though,

$$A\,\mathbf{x} = \mathbf{b}$$

$$A^HA\,\mathbf{x} = A^H\,\mathbf{b}.$$

Write

$$B = A^HA$$

and

$$\mathbf{c} = A^H\,\mathbf{b}.$$

Then

$$B \mathbf{x} = \mathbf{c}$$

and B is nonnegative definite (likewise for AA^H). Applying our method to this makes all methods converge even though

$$\lambda \ (A^H A) = \lambda \ (AA^H) = \lambda^2 (A),$$

a normally disastrous event. These realizations are Breakthrough 2.

Many other things done in BOCs are pretty, but those two are essence. Of the two, Breakthrough 1 is essential. Breakthrough 2 allows 0 (1) solutions.

SUMMARY

CONVENTIONAL ALGEBRA ON DIGITAL COMPUTERS	BIMODAL OPTICAL COMPUTERS
• SEEKS $\| \mathbf{x} - \mathbf{x}_T \| < \epsilon$	• SEEKS $\| \mathbf{b} - A \mathbf{x} \| < \epsilon$
• REQUIRES ROUGHLY O (N^3) TEMPORAL COMPLEXITY	• O (1) TEMPORAL COMPLEXITY
• ALGORITHM MATCHED TO PROBLEM	• CONSTANT ALGORITHM SUFFICES
• $\epsilon \ (\| \mathbf{x} \|) \ \alpha \ \chi \ (A)$	• $\| \mathbf{b} - A \mathbf{x} \|$ -> 0 INDEPENDENTLY OF x(A)
• $E \ (\| \mathbf{x} \|) \ \alpha \ E \ (P)$	• $\| \mathbf{b} - A \mathbf{x} \| < \epsilon$ INDEPENDENTLY OF E (P)

The highlights of this period include a laboratory demonstration of an 0 (1) time solver of even singular matrix equations and the first vigorous mathematical proof of how this works. Appendix A gives those details.

In the appendix, we show papers from optics journals and mathematics journals giving in mathematical detail the proof in illustration that these concepts are workable. In terms of applications to SDI, these might range from signal processing (where constrained linear equations lead to fast image restoration) to phased array radar (where the magnitude of jammer signals is essentially irrelevant and processor speed is independent of the number of elements in the radar). While IBM is working on the approach we developed as a possible electronic product, Nodal Systems Corporation is planning on investing tens of millions of dollars to develop this technology as an optical algebra processor. That processor would be able to operate on very large (tens of thousands in each dimension) algebraic problems and achieve high accuracy even for ill-conditioned systems at very high speed.

III. MASSIVE PARALLEL INTERCONNECTION

Before this program, what was meant by massive parallel interconnection was the connection of each element of a one dimensional optical input to each element of a one dimensional optical output. If both input and output had dimensionality N, then there were N^2 parallel, weighted optical interconnections. It was argued that this offered an advantage over electronics. The argument may well be correct for a large N, but it is not altogether certain. To achieve an indisputable advantage for optics over electronics, we sought to connect a N x N input array to a N x N output array using N^4 parallel weighted interconnections. For large N (100 to 1000), the number of parallel weighted interconnections is significantly more than can ever be accomplished with electronics. Let us try to justify that statement by considering connecting a 1000 x 1000 array of electrical signals to a 1000 x 1000 array of other electrical elements using wires. By this conceptual design, we will allow ourselves 22nd century technology. For instance, we will assume that the interconnections can be made with submicron diameter wires such that the wires plus insulators are only one micron in diameter. This means that the full set of 10^{12} wires could fit in a very small cross sectional area of only one meter by one meter. Actually, of course, that is not the case. The reason a much larger area will be required is that the wires must be scrambled and criss-cross one another. If we are very clever, perhaps we could fit that into a two meter by two meter area. The length of that bundle of wires would have to exceed the width. We can optimistically assume that the length of the interconnection bundle will be only four meters. Thus, the whole interconnection can fit in a package only two meters by two meters by four meters. SDI will not fly this in the head of a missile, but the assembly of that many wires can at least be done. Some technology, not known to us, must be used to set the resistances of the various wires (the weights). Since we don't know what that technology is, we will not explain it here. The interaction among currents in those various wires will be very severe. This will vastly increase both the delays (which will be quite variable among interconnections) and the required power. Let us again invoke future technological wonders and assume those problems can be made to vanish or be negligible. Then the sole remaining problem is simply to bond the 10^{12} submicron wires to the appropriate bonding pads. Again invoking 21st century technology, we assume a bonding machine can be made that makes 1000 perfect bonds of submicron wires to appropriate bonding pads every second. Such a bonding machine would be a great marvel indeed, but it would have to operate continuously for four years just to hook up the system. Of course, we have neglected the question of how one might check out such a system. Nevertheless, these considerations suggest that, for all practical purposes, such interconnection is impossible with electronics.

BASIC CONCEPT

Fig. 2 shows the basic concept schematically. The input is a two dimensional array of modulators (A Spatial Light Modulator or SLM). In this drawing, it is shown as a transmissive SLM. In other cases, it can reflective. The two dimensional output array can be thought of as detectors, bi-stable optical devices, or any other useful components. A lens or lens system images a two dimensional array of holograms onto the two dimensional array of outputs through the SLM. All of the holograms are simultaneously illuminated. The ij hologram is imaged onto the ij SLM. The strength of light from the ij hologram to the kl element of the SLM may be called T_{ijkl}. The amount of light transmitted through the kl element depends on the input which we may call a_{kl}. Thus the transmitted light from the ij hologram through the kl element is $T_{ijkl}\ a_{kl}$. The lens collects such contributions over the entire SLM. That is, the amount of light arriving at the ij element in the output plane is

$$b_{ij} = \sum_{kl} T_{ijkl}\ a_{kl}.$$

We recognize this as the ij element of the product of the two dimensional matrix A whose ij element is a_{ij} with the four dimensional tensor whose ijkl component is T_{ijkl}. Rewriting this in more compact form, we have

$$B = TA.$$

A very detailed analysis of the potientiality and limitations of this technique may be found among the references in Appendix B.

APPLICATIONS

Numerous applications of this technology can be found. The one developed especially for this program was massively parallel cellular array processors. This is discussed in the appendix. Numerous other applications are obvious and have begun to be discussed in the literature. Perhaps the most obvious is optical neural networks. Other applications are generalized Hough transforms and digital optical computers.

generalized Hough transforms and digital optical computers. In another portion of the overall ONR/SDI program, Peter Guilfoyle and co-workers made significant improvements in the digital optical computer concepts first described by Morozov*. If we are willing to simple pre and post processing of input data, we can generalize this technique to become a general purpose optical computer. This work has attracted world wide attention and numerous citations. In addition, both the neural network aspects and the digital optical computing aspects are being pursued at multi-million dollar levels by Nodal Systems Corporation. Again, the program has done its job of stimulating an entire new area (Appendix C).

IV. OTHER DEVELOPMENTS

An important early paper of this program was on optical Fredkin gates. Since the publication of that paper, this work has diverged into two directions. First, there has been considerable work in extending the two dimensional Fredkin gate array to three dimensions. This appears to have some real advantages over prior technology. Second, this work has led (under other sponsorship) to the realization that optics can accomplish what computer theorist have been dreaming of for last 15 years: the performance of computing operations at less than kT per operation. All of these matters are discussed in substantial detail in appendix C.

* H. E. Elion and V. N. Morozov, "Optoelectronic Switching Systems in Telecommunications and Computers," Marcel Dekker, N.Y. (1984).

There was important early work in making optical pattern recognition filters that not only had the invariance properties which are being gladly sought but also the property of being very easy to fabricate. This work has led to considerable progress. We are now at the point where immensely powerful optical pattern recognition mask can be designed and fabricated in a very simple way.

Finally, there was some preliminary work on how these concepts apply to optical neural networks.

These areas are expanded upon in Appendix D.

V. CONCLUSIONS

A variety of totally new concepts were introduced and established as feasible during the course of this contract. Each of them is a subject of intense continuing research around the world. Many of them are now being pursued commercially in America and will undoubtedly find their way into the SDI effort of our country. In addition, the massive commercial applications anticipated and funded as a result of work reported here constitute an outstanding example of the usefulness of the SDI program in creating new technology of broad general usefulness for America.

APPENDIX A

BIMODAL OPTICAL COMPUTERS

The first paper in this field (Appl. Opt. 25, 3128) was completed before the beginning of this contract. It showed (as described in the main text of our report) the severe limitations in principle on Bimodal Optical Computer (BOC) convergence.

The first work under this contract (Opt. Eng. 26, 22) showed that in many cases convergence occurred even when it could not be guaranteed.

The two key papers showed how to get convergence for all matrices (Appl. Opt. 26, 4906) and why this method works (Linear and Multilinear Algebra 25, 215). The experimental demonstration followed immediately (SPIE 936, 315).

Extending this to new algebra problems like eigen problems (SPIE 634, 86) and nonlinear algebra (SPIE 936, 309) increased the utility.

The ultimate SDI application is jam resistant high speed radar array data processing (Microwave and Optical Technology Letters 1, 236).

Bimodal optical computers

H. John Caulfield, John H. Gruninger, Jacques E. Ludman, K. Steiglitz, H. Rabitz, J. Gelfand, and E. Tsoni

Analog optical solutions of numerical problems tend to be fast, simple, and inaccurate. Digital optical or electronic solutions to the same problems tend to be slower, harder, and more accurate. In circumstances outlined here, hybrid analog-digital systems can be built which give the accuracy of digital solutions with intermediate degrees of speed and simplicity. Because at any instant these processors are working in either the analog or the digital mode, we call them bimodal optical computers.

I. Introduction

While optical digital computers have been drawing great attention,[1-7] it is only in analog computation that optics is known to excel over electronics. In this paper we offer a limited exploration of a proposed link between these two fields of optics. That is, we will discuss hybrid optical numerical processors. We seek the numerical accuracy of digital computing while still retaining some of the speed and power advantages of analog optics. To do this we must mix analog optics with digital electronics (or electrooptics or optics) to bootstrap the accuracy. We call this hybrid a bimodal optical computer.

While some of these concepts are new to optics, many are not new to science in general. Our purpose in this paper is to call the attention of optics workers to this area. We will present a general approach and then specialize to one very specific and simple problem: Linear algebraic equations. The method is clearly extendable to nonlinear problems and other linear problems.

II. Generic System

The generic system is comprised of three properly interacting systems: an optical analog solver of the basic problem; a memory; and an accurate (digital or hybrid) calculator of the solution accuracy. The basic cycle is as follows:

calculate an approximate solution with the optical analog processor;

remember that solution to high accuracy;

calculate the solution accuracy with the accurate computer;

repose the problem as an error reduction problem;

solve with an optical analog processor;

using the just-calculated improvement and the stored prior solution, calculate and remember the improved solution with the accurate computer;

calculate the solution accuracy with the accurate computer;

if the solution is accurate enough, stop;

if not, recycle.

Clearly, the convergence condition is that the error be reduced in each iteration. If this is the case, as we will show, the optical analog processor no longer limits solution accuracy.

In a purely digital system, the primary consumer of space, weight, power, time, and cost would be the solver (direct or iterative) of the problem solved by the relatively small, low-weight, power conservative, fast, and inexpensive optical analog processor. Thus there is the potential for significant overall system improvement using this hybrid approach.

There are two major forms the accurate processor can take. First, it can be a special purpose, fast, inexpensive digital processor. For reasons which will soon become evident, we call the hybrid system involving such a processor a mathematical problem solver. Second, the accurate processor could be a physical system interacting with the world. The problem is then isomorphic with the control theory. We call such a processor a physical problem solver. With a mild effort, the reader should become convinced that these two problem solvers use the same mathematics.

H. J. Caulfield is with University of Alabama in Huntsville, Center for Applied Optics, Huntsville, Alabama 35899; J. H. Gruninger is with Aerodyne Research, Inc., 45 Manning Road, Billerica, Massachusetts 01821; J. E. Ludman is with Rome Air Development Center/ES, Hanscom AFB, Massachusetts 01731; E. Tsoni is with University of Crete, Department of Computer Science, Iraklion, Crete, Greece; the other authors are with Princeton University, Princeton, New Jersey 08544.

Received 6 May 1986.
0003-6935/86/183128-04$02.00/0.

III. Accuracy Analysis

We will examine the bimodal optical computer (BOC) with specific emphasis on linear algebra as might be used, for example, for numerical solution of partial differential equations. The generic BOC method was originally proposed by Thompson[6] some time ago for iteratively improving the precision of mechanical devices which were used for the simultaneous solution of linear equations. This method appears to provide some considerable benefit for situations where a low-accuracy but fast device is available for providing approximate solutions to partial differential equations. This can then be linked to a higher accuracy device which is particularly well suited for forward substitution of the approximate solution into the original equation. The BOC iterative scheme, besides having been proposed by Lord Kelvin,[6] is a standard numerical approach to the iterative solution of linear systems and has solution of linear systems and has been analyzed with respect to numerical round-off error by Wilkinson[7] and Stewart[8] among others[9]. A working model of this analog and digital bimodal electrical computer has also been constructed by Karplus.[10] This work reexplores and extends the prior work and incorporates modern linear and nonlinear optical computer techniques.

We can summarize this idea in the following way. Suppose we want to solve the n-dimensional linear system of equations,

$$\mathbf{A}\,\mathbf{x} = \mathbf{b}. \tag{1}$$

Here A is a given matrix, **b** is a given vector, and **x** is the sought-after solution vector.

These problems are of great interest in their own right. In addition such systems with high dimensions arise when linear partial differential equations are solved by the finite difference method. Many other problems can be recast in this form. Suppose further that we have built a discrete optical analog processor for this problem which gives an approximate solution that can be summarized with the equation

$$\tilde{\mathbf{A}}\,\tilde{\mathbf{x}} = \tilde{\mathbf{b}}, \tag{2}$$

where $\tilde{A}$ and $\tilde{\mathbf{b}}$ differ from A and **b** because of the limited accuracy of the analog components. We now have an approximate solution to our problem $\tilde{x}$, which typically is accurate to a few percent. Next, we use a digital electronic computer to form the residual

$$\mathbf{r} = \mathbf{b} - \mathbf{A}\tilde{\mathbf{x}} \tag{3}$$

using the actual high-precision versions of A and **b**. Notice that this step entails only substitution of the current solution **x** in the modal equations, a relatively fast operation for even a modest digital computer. Subtracting Eq. (3) from Eq. (1) with digital electronics, we can write

$$\mathbf{A}(\mathbf{x} - \tilde{\mathbf{x}}) - \mathbf{r} = 0 \tag{4}$$

call the current solution error

$$\mathbf{x} - \tilde{\mathbf{x}} = \Delta\mathbf{x}, \tag{5}$$

and write Eq. (4) as

$$\mathbf{A}(\Delta\mathbf{x}) = \mathbf{r}. \tag{6}$$

We now have a problem of the same form as the original with A being the same matrix, except with the inhomogeneity term **b** replaced by the residual vector **r**.

We now want to use the analog optical computer again to estimate **Δx** and refine the solution $\tilde{x}$, but we first scale the equations by an appropriate number S to bring the voltages and currents back to the levels in the first solution. Thus we solve

$$\tilde{\mathbf{A}}\,y = S\mathbf{r} \tag{7}$$

and then use the estimate

$$\Delta\mathbf{x} = y/S \tag{8}$$

to refine the current solution to

$$\mathbf{x} = \tilde{\mathbf{x}} + \Delta\mathbf{x}. \tag{9}$$

This process can be iterated and in favorable conditions will converge quickly to solutions of accuracy only by the digital computer representation of A, **b**, and the digital computation of Eq. (3). The description above for the iterative procedure was given in terms of a linear equation; however, this concept may also be applied to nonlinear systems and would take advantage of the unique capacity of nonlinear analog circuits for the solution of the nonlinear algebraic equations of the discretized system. An analysis similar to the above treatment will again apply since the equations become quasi-linear near the true solution.

We might call this a floating-point analog computation where the scaling parameter S acts as a radix, varying from stage to stage with the size of the residuals in the equations. We note that this technique is quite similar to the very standard iterative numerical methods, such as Newton's method. In addition, we see that this technique marries analog and digital computers in a most congenial way—we take advantage of the speed and highly parallel nature of the analog system as well as the memory and high precision of the digital system in the external loop.

We have examined the stability and convergence properties of the iteration process for this BOC. To first order we can model the error caused by solving the system on an analog computer (Eq. 2) by [Ref. 8, Corollary (3.7)]

$$(\mathrm{A} + \mathrm{E})^{-1} = (\mathrm{I} - \mathrm{F})A^{-1}, \tag{10}$$

where E is the error in the matrix due to the analog representation. The norm of F is bounded by

$$\|\mathrm{F}\| \leq \frac{k(\mathrm{A})(\|\mathrm{E}\|/\|\mathrm{A}\|)}{1 - [k(\mathrm{A})\|\mathrm{E}\|/\|\mathrm{A}\|]}\,. \tag{11}$$

$\|\cdot\|$ is a matrix norm, and the condition number of A is defined by

$$k(A) = \|\mathrm{A}\| \cdot \|\mathrm{A}^{-1}\|. \tag{12}$$

Substituting Eq. 10 into Eq. 6 gives

$$\mathbf{x}_{k-1} = \mathbf{x}_k - \delta_k = \mathbf{x}_k - (\mathrm{I} - \mathrm{F})\mathrm{A}^{-1}(\mathbf{b} - \mathrm{A}\mathbf{x}_k). \qquad (13)$$

Letting $\mathbf{x}^* = \mathrm{A}^{-1}\mathbf{b}$ be the exact solution, we can rearrange this to yield

$$\mathbf{x}_{k-1} - \mathbf{x}^* = -F(\mathbf{x}_k - \mathbf{x}^*), \qquad (14)$$

and taking the norms of both sides,

$$\|\mathbf{x}_{k+1} - \mathbf{x}^*\| < \|\mathbf{F}\| \cdot \|\mathbf{x}_k - \mathbf{x}^*\|. \qquad (15)$$

We thus have a sufficient condition for geometric convergence of the process, namely, $\|\mathrm{F}\| < 1$, which is satisfied if

$$k(\mathrm{A})\left[\frac{\|\mathrm{E}\|}{\|\mathrm{A}\|}\right] < \frac{1}{2}, \qquad (16)$$

where $k(\mathrm{A})$ is the condition number of A, and $\|\mathrm{E}\|$ is the error in the analog representation of the true matrix A. Of course, when convergence takes place, the errors in the digital computation may ultimately overtake the effect of the analog error that is modeled here, although the effects of analog noise may prevent that kind of ultimate accuracy.

Since $\|\mathrm{E}\|/\|\mathrm{A}\|$ may be ~0.01 for optical analog processors, Eq. (16) requires that $k(\mathrm{A}) < 50$. This is quite restrictive but perhaps quite pessimistic. Simple equilibration of rows may change A to A′ with

$$k(\mathrm{A})' \ll k(\mathrm{A}).$$

Furthermore, a variety of other mathematical tricks can be performed. We can replace Eq. (9) with

$$\mathbf{x} = \bar{\mathbf{x}} + \Theta\Delta\mathbf{x} \qquad (17)$$

and seek to use the convergence factor Θ to force convergence in analogy with stochastic approximation. We can replace Eq. (6) by

$$\mathrm{A} + \mathbf{pq}^T(\Delta x) = \mathbf{r}, \qquad (18)$$

where $\mathbf{q}$ is chosen orthogonal to $\Delta\mathbf{x}$ and $\mathbf{p}$ is a free vector so that A and $\mathbf{pq}^T$ are of the same dimensionality as A. Calling

$$\mathrm{A}'' = \mathrm{A} + \mathbf{pq}^T, \qquad (19)$$

we seek $\mathbf{p}$ values to make

$$k(\mathrm{A}'') \ll k(\mathrm{A}). \qquad (20)$$

IV. Nonlinear Problems

Perhaps the most important payoff with BOCs may be associated with the solution of nonlinear problems. Many physical phenomena result in nonlinear differential or ultima*ely algebraic equations for solution. Such problems are notoriously difficult to treat by conventional numerical methods on digital computers. This comment follows since the algorithms will involve linearization or perhaps iteration with convergence being slow or perhaps nonexistent in highly nonlinear problems. A more suitable approach would be based on directly building the nonlinear behavior into the calculation process. It appears possible to construct hybrid machines based on this logic following lines parallel to that discussed in Sec. III. The key to this approach rests on the fact that nonlinear electronic or optical elements can be readily made and integrated together into an overall nonlinear computer.

As a simple example of a nonlinear problem we may consider the search for roots of a polynominal $p(x)$ in the real variable x. It is straightforward to use optical methods to evaluate polynomials via Horner's rule. Optical polynomial evaluation can be analog[11] or digital. Some tricks to accommodate dynamic range, allow root searching by scanning, extend the range of problems addressed, etc. are given in the latter reference. Root searching for real roots simply by scanning through x and watching for $p(x) = 0$ conditions is straightforward and fast. It is, however, not likely to be highly accurate. Suppose we identify an approximate real root x_0. We can then evaluate $p(x_0)$ and

$$p_1(x) = p(x) - p(x_0)$$

digitally. Assuming we are now close to the true root, we can now change the scale of both p_1 and x to gain sensitivity. We might substitute $y = 10x$ and $q_1 = 10p_1$ and then search $q_1(y)$ as before. This leads to a better approximation x_1 as can be verified by digital evaluation of $p(x_1)$. Accuracy is limited by the condition number of the polynomial because that limits the accuracy of the polynomial evaluation. Other similar examples can be found, and a general set of logic can be set forth as discussed below.

A nonlinear computer of the type discussed in the first paragraph could likely be of limited accuracy but capable of achieving an extremely rapid solution without the introduction of artificial linearization or iteration algorithms. The machine could be used alone or incorporated into an overall hybrid device along the lines discussed in Sec. III and the polynomial root searching example. This would entail introducing a high-accuracy digital computer as a means of monitoring residual errors. Updated corrections to the original fully nonlinear solution could be achieved by again using the nonlinear solution if it is close enough to the true answer that the nonlinear computer effectively operates in the linear mode after the first cycle. As an alternative it would be possible to construct an additional linearized version of the machine for the accuracy updates on the solution. These approaches may be theoretically modeled as well as demonstrated in the laboratory, and we plan to carry out such studies in the future.

V. Conclusions

Analog optics, when adequate for a task, is usually superior in speed, size, power consumption, and cost to all competitors. What we have suggested here is a means to extend the set of situations for which analog optics is adequate. Many studies remain to be performed on both algorithms and hardware. Nevertheless, the general concept of a hybrid system appears to be extremely promising.

Work sponsored primarily by the U.S. Army Research Office under contract DAAG-29-84-C-0026.

Some of H. J. Caulfield's work was sponsored under SDI/IST prime contract N00014-85-K-0479.

References

1. H. J. Caulfield, J. A. Neff and W. T. Rhodes, "Optical Computing: The Coming Revolution in Optical Signal Processing," Laser Focus/Electro-Opt. 100–106, 108, 110 (Nov. 1983).
2. D. Psaltis, Ed., "Optical Computing," Special Issue of Opt. Eng. (Jan 1984).
3. H. J. Caulfield and H. Szu, "The Mutual Time-Frequency Content of Two Signals," Proc. IEEE 72, 902 (1984).
4. J. A. Neff, Ed., "Optical Computing," Special Issue of Opt. Eng. (Jan 1985).
5. H. J. Caulfield, R. Athale, and W. C. Collins, "Optical Matrix-Matrix Multiplier Based on Outer Product Decomposition," Appl. Opt. 21, 2089 (1982); R. P. Bocker, H. J. Caulfield, and K. Bromley, "Rapid Unbiased Bipolar Incoherent Calculator Cube," Appl. Opt. 22, 804 (1983); P. S. Guilfoyle, "Rapid Unbiased Bipolar Incoherent Calculator Cube," Opt. Eng. 23, 20 (1984).
6. W. Thompson (Lord Kelvin), Proc. R. Soc. London 28, 111 (1878).
7. J. H. Wilkinson, *Rounding Errors in Algebraic Processes* (Prentice-Hall, Englewood Cliffs, NJ, 1964).
8. G. W. Stewart, *Introduction to Matrix Computations* (Academic, New York, 1973).
9. E. Tsoni, K. Steiglitz, J. Gelfand, and H. Rabitz, "Stability Second Convergence Properties of Lord Kelvin's Bimodal Computer," unpublished manuscript, Princeton U. Princeton, NJ, (Dec. 1984).
10. W. J. Karplus and R. A. Russell, "Increasing Digital Computer Efficiency with the aid of Error-Correcting Analog Subroutines," IEEE Trans. Comput. C-20, 831 (1971).
11. C. M. Verber, R. P. Kenan, H. J. Caulfield, J. E. Ludman, and D. P. Stilwell, "Pipelined Polynomial Processors Implemented with Integrated Optical Components," Appl. Opt. 23, 817 (1984).

Potter continued from page 3121

Optical monitoring of weld penetration

A system is being developed to monitor weld penetration optically and produce a signal for controlling an arc welder. The system is aimed at automatic welders, robot welders in particular. Made from small, low-cost components and utilizing optical fibers to conduct the signals, the system is immune to the electromagnetic interference that is common in industrial environments.

The monitor directs collimated light from a small diode laser at the molten pool of metal bene‑ ‑ the arc (see Fig. 16). A filter intercepts the reflected beam to suppress ‑ ‑raneous light, including light from the welding arc. A position-sensitive detector at a distance from the pool intercepts the beam reflected by the pool.

If the weld penetrates the workpiece completely, the curvature of the pool surface suddenly changes. This causes a sudden deflection of the reflected light beam, and consequently a displacement of the beam spot on the detector. Signal-processing circuitry determines the amplitude and rate of beam displacement to detect penetration and to generate control signals for the robot to regulate welding parameters.

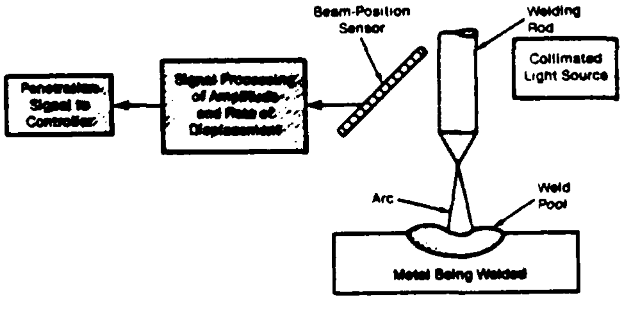

Beam-Position Sensor
Welding Rod
Collimated Light Source
Penetration Signal to Controller
Signal Processing of Amplitude and Rate of Displacement
Arc
Weld Pool
Metal Being Welded

Fig. 16. Bounding off the meniscus of a pool of molten metal, a laser beam impinges on a position-sensitive photodetector. The beam diameter can be adjusted for the width of the weld. Optical filters screen out the light from the arc.

The monitor is insensitive to changes in weld current, welder speed, and the thermal properties of the welded metal except as they affect weld penetration. The monitoring principle is adaptable to other types of welding, including tungsten/inert-gas, laser, and electron-beam techniques.

This work was done by Jonathan Maram of Rockwell International Corp. for Marshall Space Flight Center. Refer to MFS-29107.

High-flux atomic-oxygen source

A proposed apparatus can generate high fluxes (about 10^{15} atoms/cm^2-s) of ground state (^3P) oxygen atoms. The kinetic energy would be variable in the range of 3–10 eV, and the beam would be free of contaminants, such as ions, metastable ^1S or ^1D oxygen atoms, or other neutral species. Designed specifically to study the degradation of materials and spacecraft glow phenomena in low earth orbits, this oxygen-atom beam source could be used to study gas-phase collision phenomena involving energetic oxygen atoms.

In the proposed source (see Fig. 17) electrons are generated at a heated filament of LaB_6 or W. Bias voltages V_1 and V_2 accelerate the electrons to the proper energy (6.5 eV) to maximize the dissociative attachment of a beam of O_2 gas (that is, the separation of O_2 molecules into O^- ions). A solenoidal magnetic field provided by superconducting coils contains the electrons e and the ions O^- produced in the dissociative-attachment process.

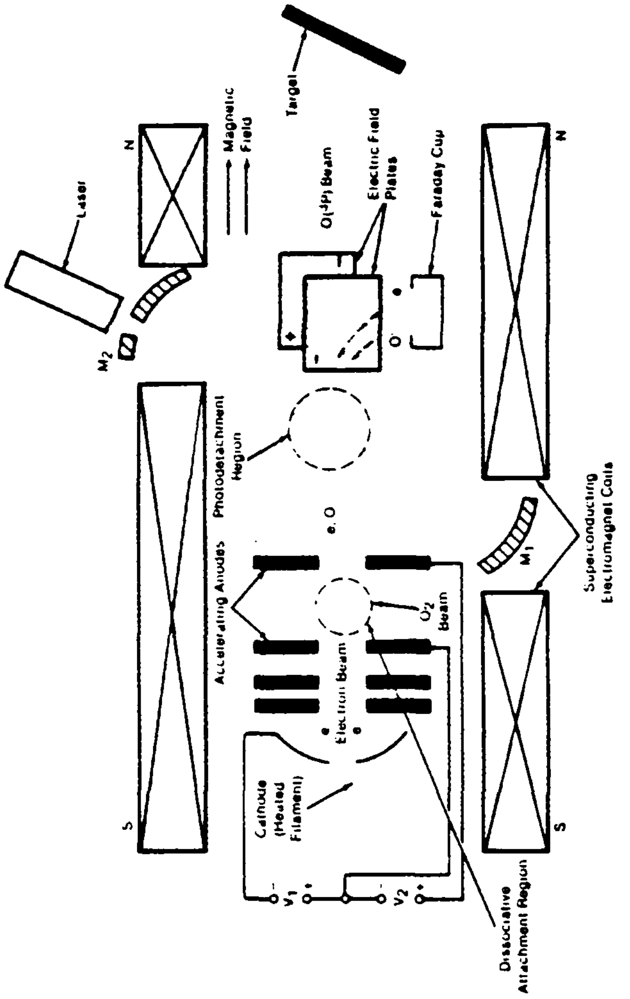


Fig. 17. Accelerated electrons strike a beam of O_2 gas in the dissociative-attachment region, producing O^- ions. The O^- ions are accelerated to the desired final energy and pass through the photodetachment region to form $O(^3P)$ atoms. These pass between electric field plates to remove O^- and e and then strike the target.

continued on page 3288

Speed and convergence of bimodal optical computers

Mustafa A. G. Abushagur, MEMBER SPIE
University of Alabama in Huntsville
Electrical and Computer Engineering
Department and
Center for Applied Optics
Huntsville, Alabama 35899

H. John Caulfield, FELLOW SPIE
Center for Applied Optics
University of Alabama in Huntsville
Huntsville, Alabama 35899

Abstract. A bimodal optical computer (BOC) for solving a system of linear equations is presented. The BOC can achieve accuracies comparable to those of the digital computer, and its speed is far superior in solving a system of linear equations. The advantage in speed increases with the size of the matrix. The problem of the convergence of the solution using the BOC is investigated. It is found that by using a BOC with an error as high as 50% in the matrix's optical mask and 1% in the electro-optical devices, convergence is achieved for matrices with condition numbers of 25. The effect of the condition number on the convergence of the solution is investigated. It is found that matrices with large condition numbers converge very slowly. Convergence for matrices with condition numbers higher than 250 was achieved. A means of improving the condition number of a matrix is also introduced.

Subject terms: optical computing and nonlinear optical signal processing; numerical processors; matrix processors; optical hybrid processors; convergence; algebra.

Optical Engineering 26(1), 022-027 (January 1987).

CONTENTS

1. INTRODUCTION

Analog optics is very attractive for signal processing and computing because of its ability to process two-dimensional data in parallel very rapidly. Unfortunately, this high speed parallel processing achieves only low accuracy because of the nature of the analog processing, especially in optical systems, where accuracy problems arise from errors in writing and reading the signals using the I/O electro-optical devices. In contrast, digital electronics is much slower but much more accurate. A compromise (hybrid) system, the bimodal optical computer, appears to be intermediate in both speed and accuracy. This method, introduced by Caulfield et al.[1] and described in Sec. 2, combines the high speed and parallelism of analog optics with the high accuracy of digital electronics using Lord Kelvin's iterative method.[2] In Sec. 3 we present a numerical analysis for the convergence of the solution of a system of linear equations. In Sec. 4 we present computer simulations of the BOC to study the dependence of the solution convergence on the condition number of the matrix and on the errors in representing the I/O data in the optical system. In Sec. 5 we compare the time required to solve a system of linear equations using the BOC to that required by the digital computer. In Sec. 6 a means of reducing the condition number of a matrix is examined, and in Sec. 7 conclusions and final remarks are drawn.

2. BIMODAL OPTICAL COMPUTER ALGORITHM

The bimodal optical computer works in the following manner for solving a system of linear equations:

$$A\mathbf{x} = \mathbf{b} \ . \tag{1}$$

where A is an n × n matrix and **x** and **b** are n × 1 vectors. A and **b** are given. The **x** is unknown and is computed as follows:

(a) Use the optical analog processor to compute an approximate solution $\mathbf{x}_0$ of the linear system. The subscript zeros indicate inaccuracies in the optics and electronics, so the system of equations solved by the optical analog processor is

$$A_0\mathbf{x}_0 = \mathbf{b}_0 \ . \tag{2}$$

(b) Store the solution $\mathbf{x}_0$ to a high accuracy with the digital computer. Use a dedicated digital processor to calculate the residue

$$\mathbf{r} = \mathbf{b} - A\mathbf{x}_0 = A(\mathbf{x} - \mathbf{x}_0) = A\Delta\mathbf{x} \ . \tag{3}$$

Invited Paper ON-107 received June 10, 1986; revised manuscript received Sept. 17, 1986; accepted for publication Sept. 17, 1986; received by Managing Editor Oct. 13, 1986.

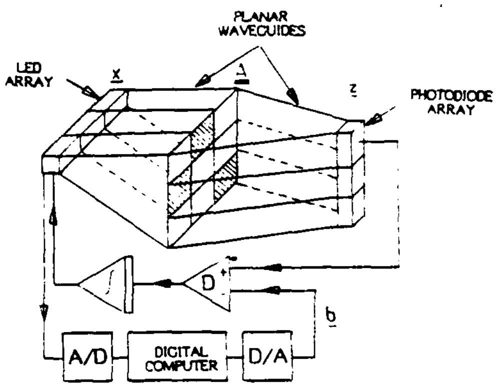


Fig. 1. System layout of the bimodal optical computer.

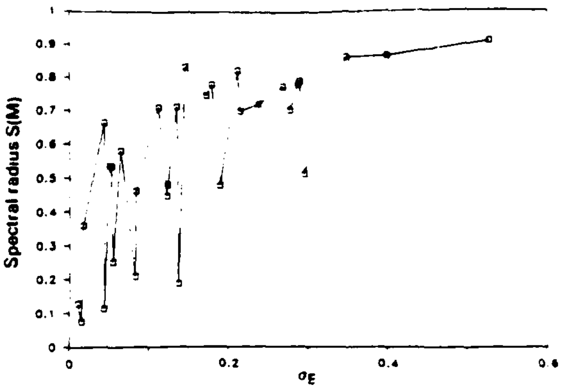


Fig. 2. Spectral radius S(M) as a function of the standard deviation of the error matrix.

(c) Use the optical analog processor to solve the new linear system for $\Delta\mathbf{x}$:

$$A_0\mathbf{y} = s\mathbf{r}_0 \ , \tag{4}$$

where $\mathbf{y} = s\Delta\mathbf{x}$ and s is a "radix," or scale factor, chosen to make good use of the dynamic range of the system.

(d) Use the digital processor to refine the solution $\mathbf{x}_0$ for $\mathbf{x}_1$:

$$\mathbf{x}_1 = \mathbf{x}_0 + \Delta\mathbf{x} \ . \tag{5}$$

If the refined solution $\mathbf{x}_1$ is accurate enough, terminate the iterations. Otherwise, return to (b), (c), and (d) for a more refined solution.

3. CONVERGENCE OF THE SOLUTION

Figure 1 is a block diagram of the BOC. The solution of the linear algebraic equation will be computed optically using the method introduced by Cheng and Caulfield.[3] The heart of the processor is the fully parallel Stanford matrix-vector multiplier.[4] Input lights representing $\mathbf{x}$ components are spread vertically onto the columns of an attenuating mask representing A. Row sums of the transmitted light are detected to give components of the output vector $\mathbf{z}$. For all k, we allow

$$\delta_k = b_k - z_k \tag{6}$$

to drive x_k. Here, z_k is a component of the calculated $\mathbf{z} = A\mathbf{x}$.

The convergence of the solution of the problem depends on two factors: the convergence of the solution of the system given in Eq. (2) by the analog processor and the convergence of the solution for the system given by Eq. (1) by the optical-hybrid processor (BOC). The convergence of the solution of Eq. (2) is discussed by Cheng and Caulfield. They report that if the matrix is a positive definite (a matrix with positive eigenvalues), then the solution will converge regardless of the size of the matrix. This simply applies to step (a) of the procedure outlined in Sec. 2.

We turn next to the total process, presenting a numerical analysis for the convergence of the solution and its dependence on the condition number of the matrix. The condition number of the matrix A is defined as

$$\chi(A) = \|A\| \cdot \|A^{-1}\| \ . \tag{7}$$

where the double bars denote the norm of the matrix. If we consider the Euclidean norm, then

$$\chi(A) = \|A\|_2 \cdot \|A^{-1}\|_2 \geq \frac{|\lambda|_{max}}{|\lambda|_{min}} \ . \tag{8}$$

where $|\lambda|_{max}$ and $|\lambda|_{min}$ are the maximum and minimum eigenvalues of the matrix A. The equality is satisfied if A is a symmetric positive definite matrix. The condition number is a measure of the accuracy of the $A\mathbf{x} = \mathbf{b}$ solutions. The larger the condition number, the less accurate the result achieved with any fixed-accuracy computer.

From Eq. (2) the solution $\mathbf{x}_0$ is given by

$$\mathbf{x}_0 = \mathrm{B}_0\mathbf{b}_0 \ . \tag{9}$$

where $\mathrm{B}_0 = (A_0)^{-1}$, and if $\mathbf{x}_i$ is the solution after the *ith* iteration, then

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \Delta\mathbf{x} \ . \tag{10}$$

where $\Delta\mathbf{x}$ is given by

$$\Delta\mathbf{x} = \mathrm{B}_0\mathbf{r} \ . \tag{11}$$

Therefore,

$$\mathbf{x}_{i+1} = (I - \mathrm{B}_0A)\mathbf{x}_i + \mathrm{B}_0\mathbf{b} \ . \tag{12}$$

where I is the identity matrix. The condition for the convergence of the solution given in Eq. (12) is that[5]

$$S(M) \leq 1 \ , \tag{13}$$

where $M = I - \mathrm{B}_0A$ and S(M) is the spectral radius of the matrix M, which is equal to the absolute value of the maximum eigenvalue $\lambda_{max}(M)$ of the matrix M. Representing the matrix A with an optical mask (a photographic film or an SLM) is the major source of the error. We need to examine how accurate this mask should be to achieve solution convergence. Let us represent the mask's matrix by

$$A_0 = A + E \ , \tag{14}$$

where E is an error matrix. For simulations, E is generated by a Gaussian random number generator with a standard deviation σ_E. In Fig 2 the spectral radius S(M) of the matrix M is plotted versus the standard deviation σ_E of the error matrix for a matrix A with a maximum coefficient of unity (any matrix can be normalized to take this form). It is clear that the spectral radius increases as σ_E increases, which slows the rate of convergence. The interesting result is that for this particular matrix, convergence is achieved even with an error matrix of >50%.

4. COMPUTER SIMULATIONS

In this section we present a computer simulation, using the BOC, of the procedure outlined in Sec. 2. In the simulation we consider matrix masks with different accuracies. We also consider the accuracy of the LEDs and photodiodes to be 1% in writing and reading the data. Here, we are interested in finding the number of iterations required for the solution to converge to a preset accuracy.

4.1. Condition number effects

The condition number is a measure of the sensitivity of the solution of Eq. (1) to any variations. In the first part of the simulation, we tested the condition number and its sensitivity to the error matrix of the optical mask. In Fig. 3 the condition number of the mask's matrix is plotted as a function of the standard deviation of the error matrix, σ_E. The maximum coefficient of the matrix A is kept equal to unity. In Fig. 3(a) we consider a matrix with a condition number equal to 60. This curve shows that with the increase of the error in representing the matrix by an optical mask, the condition number improves, except for a very few points where the error values caused the condition number to increase. In Fig. 3(b) a matrix with $\chi(A) = 300$ is considered. Here, for the entire range of σ_E, the condition number of the mask's matrix is much smaller than 300. This interesting result shows that if we start with an ill-conditioned matrix, its mask can be well-conditioned. This will help in solving problems in which the matrix is ill-conditioned.

In testing the effects of the condition number on the convergence of the solution of the system of linear equations, we used the BOC to solve the system with a 16 bit resolution. The matrices were generated randomly using Gaussian statistics. An error of 1% of the maximum coefficient of the matrix was added to generate the mask. An error of 1% also was used in reading $\mathbf{x}_0$ and in writing $\mathbf{b}_0$. In each case we computed the condition number of the generated mask's matrix. The number of iterations required for convergence of the solution was determined for each case. The iterations were terminated if they exceeded 25 and also if $|r_k|/|r_{k-1}| > 1$, which is the condition of a solution divergence. The number of iterations required for convergence of the solution with 16 bit accuracy is plotted as a function of the condition number of the mask's matrix in Fig. 4. In these calculated data points it is quite evident that the number of iterations increases with the increase of the condition number, which is a predicted result.[6] The increase of the condition number decreases the accuracy of the solution, so more iterations are needed to achieve the desired accuracy. From Fig. 4 convergence was achieved for condition numbers as high as 230, and even with $\chi(A) > 1000$ in our experiments, convergence was achieved for some cases.

4.2. Effect of the mask's error

The major limiting factor on the speed of convergence is the accuracy with which we can represent the matrix with an attenuating optical mask. In the present state of the art, accuracies of 3 to 5% are achievable. We would like to see how

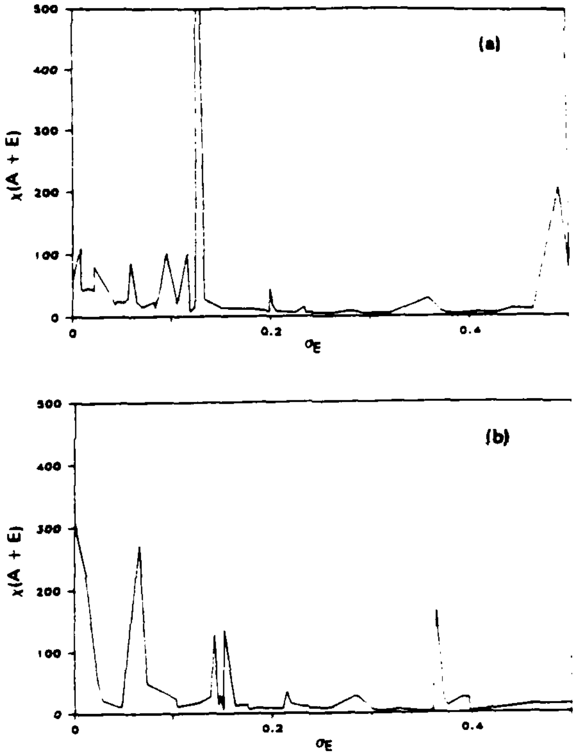


Fig. 3. Condition number $\chi(A + E)$ of the mask's matrix as a function of the standard deviation of the error matrix for (a) $\chi(A) = 60$ and (b) $\chi(A) = 300$.

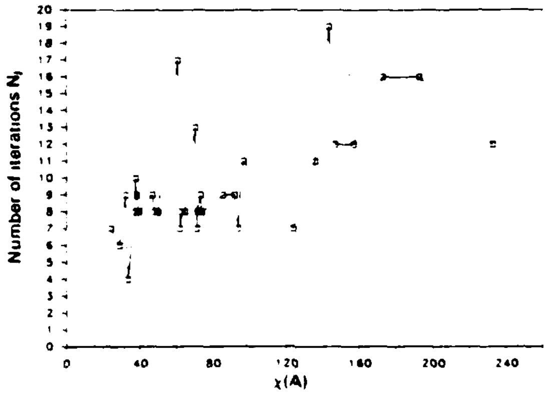


Fig. 4. Number of iterations as a function of the condition number of a randomly generated matrix A.

this will affect the speed of convergence of the solution. In this simulation we started with a matrix with a condition number $\chi(A) = 24$. Error matrices E were generated using Gaussian statistics with coefficients ranging between 1 and 55% of the maximum coefficient of the matrix A. The optical masks were generated by adding A to E. These masks were then used in the BOC to solve the system of linear equations. The number of iterations required to achieve the solution with the desired accuracy (16 bit resolution) was computed. In Fig. 5 the number of iterations is plotted as a function of the standard deviation of the error matrix, σ_E. The number of iterations increases with the increase of the error in the mask, as expected.[6] However, even with errors as high as 55% in

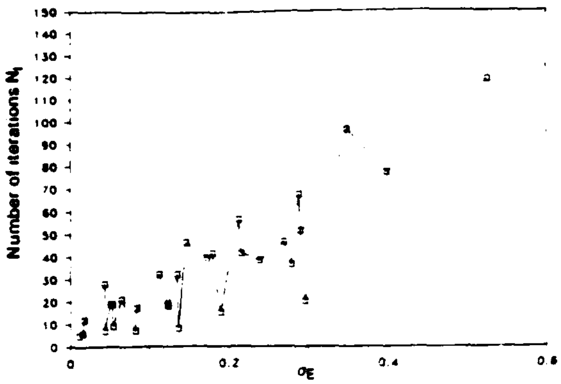


Fig. 5. Number of iterations as a function of the standard deviation of the error matrix.

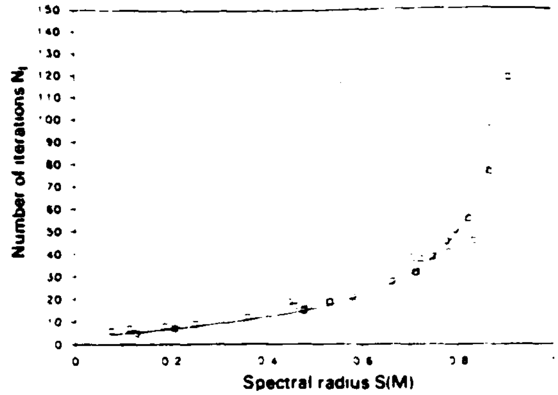


Fig. 7. Number of iterations as a function of the spectral radius. The continuous line is given by Eq. (15), and the squares are experimental data.

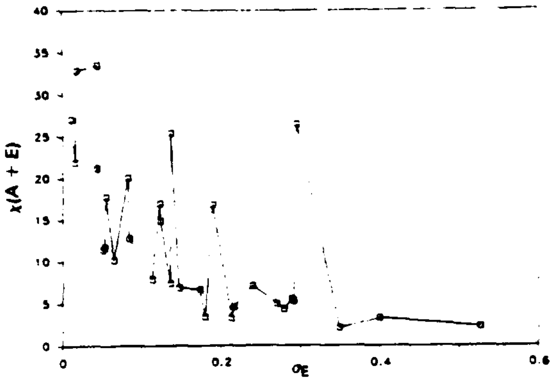


Fig. 6. Condition number χ(A + E) of the mask's matrix as a function of the standard deviation of the error matrix.

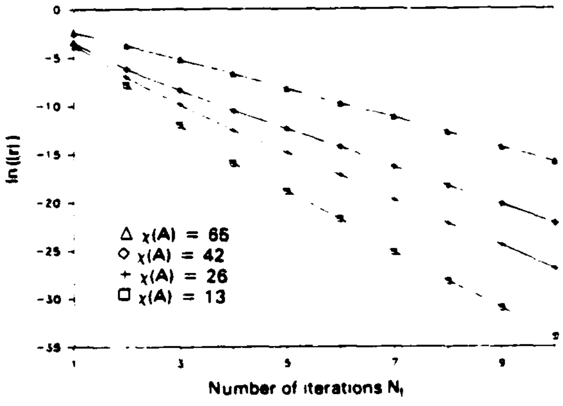


Fig. 8. Natural logarithm of residue as a function of the number of iterations for χ(A) = 13, 26, 42, and 65.

representing the matrix A with an optical mask, convergence is still retained. Of course, a larger number of iterations is required. This result is very important. It means that even with optics that are not so accurate, we can realize this optical computer that solves this class of linear algebra problems with a high accuracy and speed.

Other observations recorded in this simulation need to be highlighted. First, the condition number of the mask's matrix A + E is computed for the set of error matrices E. In Fig. 6 the condition number of the mask's matrix, χ(A + E), is plotted as a function of the error σ_E. The condition number of the mask decreases with the increase of the error almost exponentially, which is surprising since it appears to contradict the result shown in Fig. 4. We showed there that for large condition numbers we need more iterations, while here more iterations are needed for small $\chi(A_0)$. But, indeed, it is not a contradiction. Here, although these matrices have low condition numbers, they are very different from the matrix A given by the system of linear equations because of the large error involved, which makes the convergence very slow.

Second, we found in the results of the simulation that if the condition number of the mask increases for a large error, the solution will diverge, because the solution obtained in each iteration has high inaccuracies. This, in return, makes the convergence either very slow or not achievable.

Finally, we note the relationship between the spectral radius S(M) of the matrix M given by Eq. (13) and the number of iterations. We computed S(M) for each mask of Fig. 5. The number of iterations required for a solution with an error ε is given by[5]

$$N_I = -\frac{\log(\epsilon)}{R(M)}, \qquad (15)$$

where R(M) is the asymptotic rate of convergence,

$$R(M) = -\log[S(M)]. \qquad (16)$$

In Fig. 7 the number of iterations required to get a solution with a 16 bit accuracy ($\epsilon = \frac{1}{2}^{16}$) is plotted versus the spectral radius S(M). Equation (15) is plotted as a continuous line, while the data computed in the simulation are plotted as squares. The theoretical and experimental data agree well. As S(M) increases, the number of iterations increases, and as S(M) approaches unity, the convergence becomes very slow. For values of S(M) larger than unity the solution will diverge.

4.3. Rate of convergence

So far we have considered solutions with a 16 bit accuracy. We are interested in determining how many more iterations are needed to get a higher accuracy of the solution. In Fig. 8 the natural logarithm of the maximum component of the residue |r| is plotted as a function of the number of iterations for a set of matrices with different condition numbers, χ(A) = 13, 26, 42, and 65. The smaller the condition number, the higher the accuracy achieved in fewer iterations. For the condition number 13, one iteration can increase the accuracy by as many as 5 bits.

5. COMPUTATIONAL SPEED ANALYSIS

To get some quantitative values for the speed of this process compared to that of the digital computer, we calculate the number of operations required by each method, then multiply the result by the time required for each operation. We consider the total number of operations regardless of whether they are multiplications or additions.

Let us consider an n×n matrix A. The time required for one iteration of the procedure outlined in Sec. 2, T_{0I}, is given by

$$T_{0I} = T_{AI} + 2n(n+1)T_{DI} , \tag{17}$$

where T_{AI} is the time required to solve $A_0x_0 = b_0$ by analog optics and T_{DI} is the time required for one digital operation. Therefore, the time required to do I_0 iterations with the BOC is

$$T_0 = I_0[T_{AI} + 2n(n+1)T_{DI}] , \tag{18}$$

while the time required by the digital computer to solve the system of linear equations using Cholesky's method[7] in I_D iterations is

$$T_D = \left(\frac{n^3}{3} + 2n^2\right) I_D T_{DI} . \tag{19}$$

The condition we need to satisfy in order to have an advantage in time in using the BOC over the digital computer is

$$T_0 \ll T_D . \tag{20}$$

Therefore, for a clear advantage of the BOC, from Eqs. (18) to (20) we want

$$I_0[T_{AI} + 2n(n+1)T_{DI}] \ll \left(\frac{n^3}{3} + 2n^2\right) I_D T_{DI} \tag{21}$$

or

$$\kappa[T_{AI} + 2n(n+1)T_{DI}] \ll \left(\frac{n^3}{3} + 2n^2\right) T_{DI} , \tag{22}$$

where $\kappa = I_0/I_D$. Then, Eq. (22) can be rewritten in the form

$$\frac{(n^3/3) + 2n^2(1-\kappa) - 2n\kappa}{\kappa} \frac{T_{DI}}{T_{AI}} \gg 1 . \tag{23}$$

The advantage in speed in using the BOC over the digital computer is obvious from Eq. (23), and it increases as the size of the matrix n increases. To examine this condition carefully, let us rewrite Eq. (23) in the form

$$A_pA_I \gg 1 , \tag{24}$$

where

$$A_p = \frac{2[(n^3/6) + n^2(1-\kappa) - n\kappa]}{\kappa} \tag{25}$$

$$A_I = \frac{T_{DI}}{T_{AI}} . \tag{26}$$

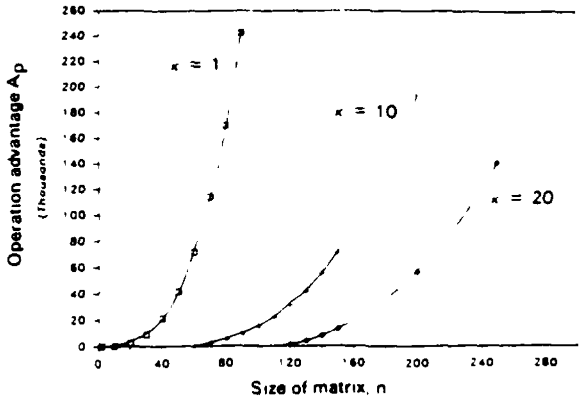


Fig. 9. Operation advantage in Eq. (25) as a function of the size of the matrix for κ = 1, 10, and 20.

Here, A_I is an "inherent advantage." A single analog operation is much faster than a digital one. The entire $Ax = b$ solution will be slower than a single digital operation, but the analog $Ax = b$ solver works at speeds independent of n. On the other hand, T_{DI} is operation dependent. It includes the time required for performing the operation and storing and retrieving the data from computer memory, which is time consuming, especially as n increases.

The factor A_p is a problem-related advantage. It is a function of the size of the matrix, n, and the ratio of the iterations, κ. The operation advantage A_p is plotted in Fig. 9 as a function of n and κ. It is clear that A_p increases rapidly as n increases, even if the number of iterations in the BOC is much larger than in the digital processor, while in reality the number of iterations of the two processes will be approximately the same for well-conditioned matrices.

6. CONDITION NUMBER REDUCTION

As mentioned earlier, the condition number is an indication of how accurate the solution of the system of linear equations will be. The larger the condition number $\chi(A)$, the more iterations are needed for solution convergence. One way of reducing the condition number of a given matrix is to normalize the matrix in the following manner:

$$a'_{ij} = \frac{a_{ij}}{(a_{i1}^2 + a_{i2}^2 + \dots\dots + a_{in}^2)^{1/2}} , \quad i = 1,2,\dots,n . \tag{27}$$

where the a_{ij} are the coefficients of the matrix A and a'_{ij} are the coefficients of the normalized matrix A_n. The ratio of the condition number of the normalized matrix A_n, $\chi(A_n)$, to that of the original matrix, $\chi(A)$, is plotted in Fig. 10. It is clear that the normalized matrix has a smaller condition number than the original matrix by a factor of approximately 0.8. We expect this to decrease the number of iterations substantially.

7. CONCLUSIONS

The speed, accuracy, and convergence of the bimodal optical computer are discussed. The BOC is similar to the digital computer in its accuracy but is faster in solving a system of linear equations than the digital computer. The speed advantage increases with the increase of the size of the matrix, which makes it a more attractive computer. The convergence of the solution as a function of the condition number and the error in

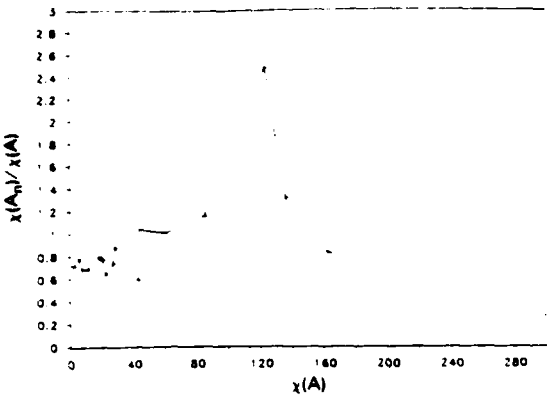


Fig. 10. Ratio of the condition number of the normalized matrix to that of the original matrix as a function of the condition number of the original matrix.

the I/O devices is also analyzed. It is found that solutions converge even for about 50% errors in the representation of the matrix by an optical mask. Although this error will reduce the speed, it will not lead to a sacrifice in the accuracy of the solution. Thus, even with today's inaccurate analog optics, we can have a powerful computer to solve this class of linear algebra problems. Normalization of the matrix will reduce its condition number, which will lead to a faster convergence.

The BOC is capable of solving other problems, both linear and nonlinear, in addition to the system represented by Eq. (1).

8. ACKNOWLEDGMENTS

This research was supported by the Innovative Science and Technology Office of the Strategic De:ense Initiative Organization, administered through the Office of Naval Research under contract N00014-85-K-6479.

9. REFERENCES

1. H. J. Caulfield, J. H. Gruninger, J. E. Ludman, K. Steiglitz, H. Rabitz, J. Gelfand, and E. Tsoni, "Bimodal optical computers," Appl. Opt. 25(18), 3123-3131 (1986).
2. W. Thompson (Lord Kelvin), Proc. R. Soc. London 28, 111 (1878).
3. W. K. Cheng and H. J. Caulfield, "Fully parallel relaxation algebraic operations for optical computers," Opt. Commun. 43, 251 (1982).
4. J. W. Goodman, A. R. Dias, and L. M. Woody, "Fully parallel, high-speed incoherent optical method for performing discrete Fourier transforms," Opt. Lett. 2, 1 (1978).
5. David M. Young, *Iterative Solution of Large Linear Systems*, Academic Press, New York (1971).
6. H. J. Caulfield and M. A. G. Abushagur, "Hybrid analog-digital algebra processors," to be published in *Optical and Hybrid Computing*, Vol. II of the SPIE Institute Series, H. H. Szu, ed., Proc. SPIE 634 (1986).
7. G. W. Stewart, *Introduction to Matrix Computations*, Academic Press, New York (1973).

Mustafa A. G. Abushagur received the B Sc. degree in 1974 from Tripoli University and M.Sc. and Ph.D. degrees from the California Institute of Technology, all in electrical engineering. He was a research assistant at the Institute of Optics, University of Rochester, while working on his Ph.D. thesis and in 1984 became an assistant professor in the Electrical Engineering Department at that university. Since January 1985, he has been an assistant professor in the Department of Electrical and Computer Engineering at the University of Alabama in Huntsville, where he is also associated with the Center for Applied Optics. His research is in the areas of coherent optics, optical computing, optical information processing, scattering, and diffraction theory. Dr. Abushagur is a member of IEEE, OSA, and SPIE.

H. John Caulfield is director of the Center for Applied Optics at the University of Alabama in Huntsville. Formerly the editor of this journal and an associate editor of several others, Dr. Caulfield is best known for his work in holography, coherent optics, and optical computing. He is the author of more than 100 refereed papers, two books, and perhaps a dozen book chapters and holds about 15 patents. A Tennessee farmer, Dr. Caulfield is a Fellow of OSA and SPIE.

Reprinted from Applied Optics, Vol. 26, page 4906, December 1, 1987

Superconvergence of hybrid optoelectronic processors

Mustafa A. G. Abushagur, H. John Caulfield, Peter M. Gibson, and Mohammad Habli

All authors are with University of Alabama in Huntsville, Huntsville, Alabama 35899; H. J. Caulfield is in the Center for Applied Optics, P. M. Gibson is in the Mathematics & Statistics Department, and the other authors are in the Electrical & Computer Engineering Department.

Received 26 August 1987.

Sponsored by Jacques E. Ludman, Rome Air Development Center.

0003-6935/87/234906-02$02.00/0.

Beginning with Lord Kelvin[1] and continuing through recent work of ours,[2-5] researchers have been interested in ways to use a fast low-accuracy processor and a slow high-accuracy processor together with intermediate but high speed. We showed earlier[2] that to guarantee convergence with a fast processor of accuracy ϵ (expressed as rms relative error) on an $A\mathbf{x} = \mathbf{b}$ problem in which the condition number is $\chi(A)$, the requirement (in approximate form) is

$$\chi(A)\cdot\epsilon < 1/2. \tag{1}$$

For analog optical algebra processors, $\epsilon = 1/50$ (2%) is excellent. This means we may expect some failures of the process for $\chi(A) > 25$. Since many matrices have much higher condition numbers, this is a severe restriction. In subsequent publications[3-5] we showed that convergence was achievable for many matrices with $\chi(A) \gg 1/(2\epsilon)$ and that A can be preconditioned (rearranged without changing its meaning) to improve $\chi(A)$. These steps brought the resulting bimodal optical computer (BOC) to the point where it was practical for some real but restricted situations.

Our goals in this Communication are twofold. First, we wish to remove some restrictions on the condition number and thus achieve convergence for a wide range of problems. Second, we wish to remove the restriction we imposed on the $A\mathbf{x} = \mathbf{b}$ solver by limiting its convergence to only positive definite matrix A and thus guarantee convergence for other matrices by modifying the algorithm.

Although the BOC can be applied to all linear algebra problems we pick the general $A\mathbf{x} = \mathbf{b}$ problem for illustration. We review here the basic ideas:

(1) Solve $A\mathbf{x} = \mathbf{b}$ optically to get $\mathbf{x}_0$.

(2) With specialized digital processor, evaluate to high accuracy

$$\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0. \tag{2}$$

(3) Normalize $\mathbf{r}_0$ to keep solutions in range.

(4) Solve optically

$$A(\Delta\mathbf{x}_0) = \mathbf{r}_0. \tag{3}$$

(5) Evaluate digitally

$$\mathbf{x}_1 = \mathbf{x}_0 + \Delta\mathbf{x}_0 \tag{4}$$

$$\mathbf{r}_1 = \mathbf{b} - A\mathbf{x}_1 \tag{5}$$

(6) If $\|\mathbf{r}_1\|$ is small enough, stop. Otherwise, go to (3) and recycle.

In the optical steps, replace A with a new matrix A_0 derived from A by adding noise to it.

$$A_0 = A + E, \tag{6}$$

where E is an error matrix generated using Gaussian statistics with a standard deviation σ_E. The new matrix will have a much better condition number, especially for an ill-conditioned starting matrix A. The digital correction steps keep the solution headed toward $A\mathbf{x} = \mathbf{b}$ not $A_0\mathbf{x} = \mathbf{b}$. In analog processors adding E is automatic because of system noise. We treat σ_E hereafter as the standard deviation of system noise.

The method proved capable of solving systems of linear equations with a wide range of condition numbers. The convergence to the solution is very rapid for small condition numbers and very large condition numbers (near singular and singular) but not as rapid in intermediate values of condition numbers. And it works best for singular, underdetermined, or overdetermined systems.

In Table I the number of iterations N_I required for a convergence to a solution with 16-bit accuracy is tabulated as a function of the error involved in the calculations. These results are obtained using a computer simulation of the bimodal optical computer. The errors considered in the calculation are defined as follows:

σ_E ≡ the standard deviation of the error matrix E;

σ_b ≡ the standard deviation of the error in writing the vector **b**, and

σ_x ≡ the standard deviation in reading **x**.

The results in Table I are for a singular matrix A. The matrices considered here are 10 × 10 and have rank of 9 and 1, respectively. When an error is added to the vector **b** the solution diverges, but by adding an error matrix E to A the solution converges very rapidly, as shown in the table. This is true for different values of the error σ_b. The technique does work even with a processor with errors larger than that shown in Table I. Also we have considered a set of Hilbert matrices,[6] which are very ill-conditioned, and their condition number increases very rapidly by the increase of size. These are used as test matrices for our new technique which was able to achieve convergence very rapidly, especially when there are errors in the vector **b**.

Thus this technique makes the solution converge for a system of equations which cannot be solved with ordinary

Table I. Convergence Behavior of 10 × 10 Matrices of Ranks 9 and 1 for Various Additive Errors

			10 × 10		10 × 10	
			Rank = 9		Rank = 1	
σ_E	σ_b	σ_x	N_I	$\|\mathbf{r}_2\|/\|\mathbf{r}_1\|$	N_I	$\|\mathbf{r}_2\|/\|\mathbf{r}_1\|$
0	0	0	1		1	
0	1.E-6	0	4	0.5585	*D*	5.1*E* + 26
1.*E*-6	1.*E*-6	0	1	3.48-6	1	8.9*E* − 7
0	1.*E*-6	1.*E*-6	*D*	2.3273	*D*	6.7*E* + 26
1.*E*-6	1.*E*-6	1.*E*-6	1	9.24-6	1	1.2*E* − 5
0	0.01	0	*D*	1.000	*D*	2.4*E* + 64
1.*E*-6	0.01	0	4	2.2*E*-2	2	2.9*E* − 3
0	0.01	0.01	*D*	1.1953	*D*	1.2*E* + 31
1.*E*-6	0.01	0.01	9	2.1*E*-1	5	6.9*E* − 2
1.*E*-4	0.01	0.01	7	1.7*E*-1	4	3.6*E* − 3
0.01	0.01	0.01	8	2.5*E*-1	5	1.3*E* − 1
0.05	0.01	0.01	11	2.4*E*-1	4	2.0*E* − 1
0.10	0.01	0.01	18	7.2*E*−1	4	8.6*E* − 2

N_I is the number of iterations required to achieve $\|r_{NI}\| = 0$ to within 16 bits. The ratio of $\|\mathbf{r}_2\|/\|\mathbf{r}_1\|$ gives another measure of convergence (or divergence indicated by *D*) rate. We have used the infinity norm for convenience.

techniques. We have tested this technique for matrices with sizes *N* up to 12 × 12 and with different ranks from (*N* − 1) to 1. For all these cases it does work with a high speed of convergence.

In previous papers we showed that the parallel analog processor proposed for the BOC is capable of solving only systems of linear equation with positive definite matrices. Here we will -ase this restriction. In general the matrix A has complex eigenvalues. If the matrix A is multiplied by the Hermetian matrix A^H, the matrix A^HA will have a non-negative eigenvalue. Now multiply the system $A\mathbf{x} = \mathbf{b}$ by A^H; then

$$A^HA\mathbf{x} = A^H\mathbf{b}. \tag{7}$$

Equation (7) gives a new system of linear equation with a non-negative eigenvalue. In practice adding E to A^HA results in a positive definite matrix. Thus general systems can be solved in this manor. Replacing A by A^HA increases the condition number (squaring at the most) but does not prevent convergence to an accurate solution of Eq. (7). Equation (7) is, of course, not well posed as $A\mathbf{x} = \mathbf{b}$. Therefore, we must use the residual of Eq. (2) not the residual of Eq. (7).

With this method most problems behave as simple problems: they converge and converge rapidly. This has been applied successfully to $A\mathbf{x} = \mathbf{b}$ problems, which are determined, underdetermined (linear programming), or overdetermined.

The method is purely algebraic and is, therefore, simply an improved approach to some numerical algebra problems. The fact that it is suited for BOC use is an independent fact.

If we can achieve fast convergence almost independently of condition number, the first practical application may be to phased array antennas where the phasing problems are linear algebra and the primary difficulty is the presence of jammers: malicious means to increase the condition number.

This research was supported by the Innovative Science and Technology Office of the Strategic Defense Initiative Organization, administered through the Office of Naval Research under contract N00014-86-K-0591.

References

1. W. Thompson (Lord Kelvin), "On the Machine for the Solution of Simultaneous Linear Equations," Proc. R. Soc. London **28**, 111 (1878).
2. H. J. Caulfield, J. H. Gruninger, J. E. Ludman, K. Steiglitz, H. Rabitz, J. Gelfand, and E. Tsoni, "Bimodal Optical Computers," Appl. Opt. **25**, 3123 (1986).
3. M. A. G. Abushagur and H. J. Caulfield, "Speed and Convergence of Bimodal Optical Computers," Opt. Eng. **26**, *022 (1987)*.
4. H. J. Caulfield, and M. A. G. Abushagur, "Hybrid Analog-Digital Algebra Processors," Proc. Soc. Photo-Opt. Instrum. Eng. **634**, 86 (1986).
5. M. A. G. Abushagur and H. J. Caulfield, "Highly Precise Optical-Hybrid Matrix Processor," Proc. Soc. Photo-Opt. Instrum. Eng. **639**, 63 (1986).
6. D. I. Steinberg, *Computational Matrix Algebra* (McGraw-Hill, New York, 1974).

Linear and Multilinear Algebra, 1989, Vol. 25, pp. 215–218
Reprints available directly from the publisher
Photocopying permitted by license only

Printed in the United States of America

On an Iterative Method for Consistent Linear Systems*

PETER M. GIBSON

Mathematics and Statistics Department and Center for Applied Optics, University of Alabama in Huntsville, Huntsville, AL 35899

(*Received September 12, 1988; in final form February 6, 1989*)

Let $Ax = b$ be a consistent linear system with A an $n \times n$ complex matrix. Suppose that G is a nonsingular $n \times n$ complex matrix for which $A + G$ is nonsingular and zero is not a multiple root of the minimum polynomial of $G^{-1}A$. It is shown that there exists a positive real number ρ such that whenever γ is a complex number with $0 < |\gamma| < \rho$ the sequence $x_0, x_1, x_2, \ldots$ converges to a solution of $Ax = b$ for every initial vector x_0, where $(A + \gamma G)x_i = \gamma G x_{i-1} + b$ for $i = 1, 2, \ldots$. Related questions are also considered.

In this note theoretical results are presented that help explain the observed behavior of a standard iterative process for solving a linear system of equations.

Analog optics is very attractive for performing matrix computations because of its ability to process two-dimensional data in parallel very rapidly [5]. Unfortunately, this high speed processing achieves only low accuracy. In contrast, digital electronic processors are slower but much more accurate. It was recently suggested [4] that linear systems can be solved iteratively by a method that combines the speed of analog optics with the accuracy of digital electronics. The proposed method is based on the usual iterative refinement of approximate solutions of linear systems (for example, see [6]). To solve a system $Ax = b$, where A is an $n \times n$ matrix, use an optical analog processor to find an approximate solution $\hat{x}$ of $Ax = b$.

1. Use a digital electronic processor to compute $r = b - A\hat{x}$.
2. Use the optical processor to find an approximate solution $\hat{e}$ to the system $Ae = r$.
3. Use the digital processor to refine the approximate solution of $Ax = b$ to $\tilde{x} = \hat{x} + \hat{e}$. If $\tilde{x}$ is accurate enough, terminate the iterations; otherwise, set $\hat{x} = \tilde{x}$ and return to step 1.

Due to inaccuracies in writing and reading the signals using electro-optical devices, the optical processor solution $\hat{x}$ of $Ax = b$ is the exact solution of a perturbed system $(A + E)x = b$, where we assume that the matrix $A + E$ is nonsingular. In the current state of the art in optical processing, the magnitudes of the entries of the error matrix E may be from three to five percent of the maximum magnitude of the entries of A.

* This work was supported by the Innovative Science and Technology Office of the Strategic Defense Initiative Organization, administered through the Office of Naval Research under contract N00014-86-K-0591.

In numerical experimentation on a digital electronic computer, the error matrices were randomly generated [1], [2]. In extensive experiments, it was found that $A+E$ would be nonsingular and that the approximate solutions converged to a solution of $Ax=b$. Application of the method can be viewed as a preconditioning of the system. It was applied to systems for which A was nonsingular with condition number varying from small to quite large [1], [4]. Later consistent systems with singular coefficient matrices were solved, and it was found that convergence to a solution of $Ax=b$ was in practice even better than for A nonsingular [2].

When the method is applied, a sequence $x_0, x_1, x_2, \ldots$ of approximate solutions to the consistent system $Ax=b$ are generated where

$$(A+E)x_i = Ex_{i-1}+b, \qquad i=1,2,\ldots.$$

It is well known (e.g. see [3]) that for *nonsingular* $A+E$ *iteration converges* to a solution of $Ax=b$ for every initial vector x_0 if and only if $\lim_{m\to\infty}((A+E)^{-1}E)^m$ exists. If this limit exists, for $n\times n$ complex matrices A and E, then we say that E is an *acceptable error matrix for* A. Let $H_n=(h_{ij})$ be the $n\times n$ matrix with $h_{i,i+1}=1$ for $i=1,2,\ldots,n-1$, and all other entries zero.

THEOREM 1 *Let A and G be $n\times n$ complex matrices with G and $A+G$ nonsingular.*

(a) *If zero is not a multiple root of the minimum polynomial of $G^{-1}A$, then there exists a positive real number ρ such that, for all complex numbers γ with $0<|\gamma|<\rho$, $E=\gamma G$ is an acceptable error matrix for A.*

(b) *If there exists a nonzero complex number γ such that $E=\gamma G$ is an acceptable error matrix for A, then zero is not a multiple root of the minimum polynomial of $G^{-1}A$.*

Proof Let A have rank r, and suppose that zero is not a multiple root of the minimum polynomial of $G^{-1}A$. Then there exist a nonsingular $n\times n$ matrix P and a nonsingular $r\times r$ matrix Q such that

$$P^{-1}G^{-1}AP=\begin{bmatrix}0 & 0\\ 0 & Q\end{bmatrix}.$$

Let

$$\rho=(\min\{|\lambda|: \lambda \text{ is an eigenvalue of } Q\})/2.$$

Then $\rho>0$. Let γ be a complex number with $0<|\gamma|<\rho$. It follows that γG is nonsingular and

$$P^{-1}((\gamma G)^{-1}A+I)P=\begin{bmatrix}I & 0\\ 0 & I+\gamma^{-1}Q\end{bmatrix}.$$

Let λ be an eigenvalue of $I+\gamma^{-1}Q$. Then $\lambda=1+\mu/\gamma$ for some eigenvalue μ of Q. Thus $|\lambda|\geqslant|\mu/\gamma|-1>2-1=1$. Therefore, $I+\gamma^{-1}Q$ is nonsingular and $|\lambda|<1$ for each eigenvalue λ of $(I+\gamma^{-1}Q)^{-1}$. It then follows that $(\gamma G)^{-1}A+I$ is nonsingular and that $\lim_{m\to\infty}(((\gamma G)^{-1}A+I)^{-1})^m$ exists. Moreover, since $(\gamma G)^{-1}A+I=(\gamma G)^{-1}(A+\gamma G)$, we see that $A + \gamma G$ is also nonsingular with

$$(A + \gamma G)^{-1}\gamma G = ((\gamma G)^{-1}(A + \gamma G))^{-1} = ((\gamma G)^{-1}A + I)^{-1}.$$

Thus $E = \gamma G$ is an acceptable error matrix for A. Therefore part (a) holds.

Now suppose that there exists a nonzero complex number γ such that $E = \gamma G$ is an acceptable error matrix for A, and that zero is a root of the minimum polynomial of $G^{-1}A$ of multiplicity k, where $k > 1$. From the Jordan canonical form of $G^{-1}A$, we see that there exists a nonsingular matrix P such that

$$P^{-1}G^{-1}AP = \begin{bmatrix} H_k & 0 \\ 0 & Q \end{bmatrix}$$

where Q is some $(n-k) \times (n-k)$ matrix. It follows that

$$P^{-1}(A + \gamma G)^{-1}\gamma GP = \begin{bmatrix} (I + \gamma^{-1}H_k)^{-1} & 0 \\ 0 & (I + \gamma^{-1}Q)^{-1} \end{bmatrix}.$$

Since $k > 1$, we see that $\lim\limits_{m \to \infty} ((I + \gamma^{-1}H_k)^{-1})^m$ does not exist, and thus $\lim\limits_{m \to \infty} ((A + \gamma G)^{-1}\gamma G)^m$ does not exist. This contradiction establishes part (b).

If A is given and E is randomly generated, one would expect E and $A + E$ to be nonsingular, and that zero would not be a multiple root of the minimum polynomial of $E^{-1}A$. Therefore, if the entries of E are chosen with magnitudes fairly small in comparison with the maximum magnitude of the entries of A, Theorem 1 indicates that E will probably be an acceptable error matrix for A.

Part (a) of Theorem 1 clearly implies the following.

COROLLARY 2 *Let A and G be nonsingular $n \times n$ complex matrices for which $A + G$ is nonsingular. Then there exists a positive real number ρ such that, for all complex numbers γ with $0 < |\gamma| < \rho$, $E = \gamma G$ is an acceptable error matrix for A.*

For nonzero singular matrices A we have the following.

THEOREM 3 *Let A be an $n \times n$ complex matrix of rank r where $0 < r < n$. Then there exists a nonsingular $n \times n$ complex matrix G such that $A + G$ is nonsingular and for each nonzero complex number γ, γG is not an acceptable error matrix for A.*

Proof Let $d = n - r$, let $n_1, n_2, \ldots, n_d$ be positive integers with $n_1 + n_2 + \cdots + n_d = n$, and define a block-diagonal matrix J by letting

$$J = \operatorname{diag}[H_{n_1}, H_{n_2}, \ldots, H_{n_d}].$$

Since A and J have the same rank, there exist nonsingular matrices P and Q such that $QAP = J$. Let $G = (PQ)^{-1}$, and let γ be nonzero complex number. It follows that $A + \gamma G$ is nonsingular with

$$(A + \gamma G)^{-1}\gamma G = ((\gamma G)^{-1}(A + \gamma G))^{-1} = P(I + \gamma^{-1}J)^{-1}P^{-1}.$$

Since $n_i > 1$ for some $i = 1, 2, \ldots, d$, we see that $\lim\limits_{m \to \infty} ((I + \gamma^{-1}H_{n_i})^{-1})^m$ does not exist, and thus $E = \gamma G$ is not an acceptable error matrix for A.

The acceptable error matrices that we have presented for a given matrix A would appear to have entries with small magnitudes in comparison with the maximum magnitude of the entries of A. However, the next two theorems show that each $n \times n$ complex matrix A has acceptable error matrices with entries of arbitrarily large magnitude. For nonsingular A, it is easy to prove the following.

THEOREM 4 *Let A be a nonsingular complex matrix. If γ is any complex number with positive real part, then $E = \gamma A$ is an acceptable error matrix for A.*

Now let $K_n = (k_{ij})$ be the $n \times n$ matrix with $k_{n1} = 1$ and all other entries zero. For nonzero scalars γ, we see that the matrix $H_n + \gamma K_n$ is nonsingular with inverse equal to the transpose of $H_n + \gamma^{-1} K_n$. We shall use this in proving the following.

THEOREM 5 *Let A be a singular $n \times n$ complex matrix of rank r, and let $d = n - r$. Then there exist d linearly independent $n \times n$ complex matrices $E_1, E_2, \ldots, E_d$ such that for all nonzero complex numbers $\gamma_1, \gamma_2, \ldots, \gamma_d$, $E = \gamma_1 E_1 + \gamma_2 E_2 + \cdots + \gamma_d E_d$ is an acceptable error matrix for A.*

Proof There exist positive integers $n_1, n_2, \ldots, n_d$ with $n_1 + n_2 + \cdots + n_d \leqslant n$ and nonsingular matrices P and Q such that

$$P^{-1}AP = \operatorname{diag}[H_{n_1}, H_{n_2}, \ldots, H_{n_d}, Q].$$

For $i = 1, 2, \ldots, d$, let

$$E_i = P \operatorname{diag}[\delta_{i1} K_{n_1}, \delta_{i2} K_{n_2}, \ldots, \delta_{id} K_{n_d}, 0] P^{-1},$$

where δ_{ij} is the Kronecker delta. Clearly, $E_1, E_2, \ldots, E_d$ are linearly independent. Let $\gamma_1, \gamma_2, \ldots, \gamma_d$ be nonzero complex numbers, and let $E = \gamma_1 E_1 + \gamma_2 E_2 + \cdots + \gamma_d E_d$. For $i = 1, 2, \ldots, d$, we have

$$(H_{n_i} + \gamma_i K_{n_i})^{-1} \gamma_i K_{n_i} = (H_{n_i} + \gamma^{-1} K_{n_i})^t \gamma_i K_{n_i} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}.$$

Therefore, $\lim_{m \to \infty} ((H_{n_i} + \gamma_i K_{n_i})^{-1} \gamma_i K_{n_i})^m$ exists for $i = 1, 2, \ldots, d$, and it follows that $A + E$ is nonsingular and that $\lim_{m \to \infty} ((A + E)^{-1} E)^m$ exists. Thus E is an acceptable error matrix for A.

References

[1] M. A. G. Abushagur and H. J. Caulfield, Speed and convergence of bimodal optical computers, *Optical Engineering* **26** (1987), 22–27.

[2] M. A. G. Abushagur, H. J. Caulfield, P. M. Gibson and M. Habli, Superconvergence of hybrid opto-electronic processors, *Applied Optics* **26** (1987), 4906–4907.

[3] A. Berman and R. J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*, Academic Press, New York, 1979.

[4] H. J. Caulfield, J. H. Gruninger, J. E. Ludman, K. Steiglitz, H. Rabitz, J. Gelfand and E. Tsoni, Bimodal optical computers, *Applied Optics* **25** (1986), 3123–3131.

[5] J. W. Goodman, A. R. Dias and L. M. Woody, Fully parallel, highspeed incoherent optical method for performing discrete Fourier transforms, *Optics Letters* **2** (1978), 1–3.

[6] G. W. Stewart, *Introduction to Matrix Computations*, Academic Press, New York, 1973.

HYBRID OPTOELECTRONIC NONLINEAR ALGEBRA PROCESSOR

Mustafa A. G. Abushagur
Electrical and Computer Engineering Department

H. John Caulfield
Center for Applied Optics
University of Alabama in Huntsville
Huntsville, Alabama 35899

ABSTRACT

A novel system for solving systems of nonlinear equations is proposed. Two different algorithms are introduced. A speed analysis of the two different algorithms is presented and compared with the speed of their digital computer counter parts. A great advantage in speed is shown for large size problems.

1. INTRODUCTION

Systems of nonlinear equations arise in the process of solving many physical problems. They are a very important class of mathematical problems. Iterative methods are used to solve such problems.

In this paper we propose a new method for solving this class of nonlinear problems using optical processors. In Section 2 the iterative methods used in solving nonlinear systems of equation is reviewed. In Section 3 the optical implementation is proposed using two different algorithms. The speed analysis of the two algorithms is given in Section 4. In Section 5 conclusions and final remarks are drawn.

2. NEWTON's METHOD

Systems of linear equations are given as follows

$$A\vec{x} = \vec{b} \qquad (1)$$

where A is an $n \times n$ matrix, $\vec{x}$ and $\vec{b}$ are $n \times 1$ vectors. In these systems A and $\vec{b}$ are given and $\vec{x}$ the solution of the system is unknown.

Nonlinear systems of equations can be represented by

$$\vec{f}(\vec{x}) = 0 \qquad (2)$$

or

$$\begin{bmatrix} f_1(x_1 \cdots x_n) \\ \cdots\cdots\cdots\cdots \\ f_n(x_1 \cdots x_n) \end{bmatrix} = 0,$$

where f_i's are nonlinear functions of $\vec{x}$

One of the methods used in solving for $\vec{x}$ in the nonlinear system of equations is Newton's method. For a single nonlinear equation, an initial solution, x_o, of the equation is assumed. and the $(k+1)^{th}$ iteration of the solution is given by[1]

$$x_{k+1} = x_k - (f_k')^{-1} f_k \tag{3}$$

where

$$f_k = f(x_k)\ ,\ \text{and}\ f_k' = \left.\frac{\partial f(x)}{\partial x}\right|_{x = x_k} \tag{4}$$

For a system of nonlinear equations, Eq. (3) can be rewritten as

$$\vec{x}_{k+1} = \vec{x}_k - (J_k)^{-1} \vec{f}_k\ , \tag{5}$$

where

$$(J)_{i,j} = \frac{\partial f_i(\vec{x})}{\partial x_j}\ , \tag{6}$$

and J is the Jacobian matrix.

Let

$$(J_k)^{-1} \vec{f}_k = \vec{c}_k\ , \tag{7}$$

then

$$J_k \vec{c}_k = \vec{f}_k\ . \tag{8}$$

Eq. (8) is a system of linear equations to be solved for $\vec{c}_k$, which is the correction needed for the $(k+1)^{th}$ solution iteration. The algorithm for solving the system of nonlinear equations will be as follows:

i) Assume a solution $\vec{x}_o$.

ii) Compute the nx1 vector $\vec{f}_k$ and the nxn matrix J_k

iii) Solve the linear system of equations $J_k\vec{c}_k = \vec{f}_k$ for $\vec{c}_k$

iv) Compute the refined solution $\vec{x}_{k+1} = \vec{x}_k - \vec{c}_k$

v) Check if the norm $\|\vec{f}_{k+1} - \vec{f}_k\| < \epsilon$ stop, otherwise go back to step (ii). ϵ is the allowable error.

3. OPTICAL IMPLEMENTATION

The iterative algorithm introduced in Section 2 requires $0(n^3)$ number of operations when used with conventional digital computer. The most expensive part of the algorithm is step (iii) to solve a system of linear equations. In previous publications[2-4] we have proposed and analyzed a hybrid optoelectronic processor, the Bimodal Optical Computer BOC, capable of solving linear systems of equations accurately and rapidly. In this section we modify that system to be used to solve systems of nonlinear equations as shown in Fig. 1. We propose two different algorithms, the first utilizes the use of the analog processor to solve the system of equations (8) approximately, and the second to use the BOC to solve the system of equations (8) exactly (within the specified accuracy).

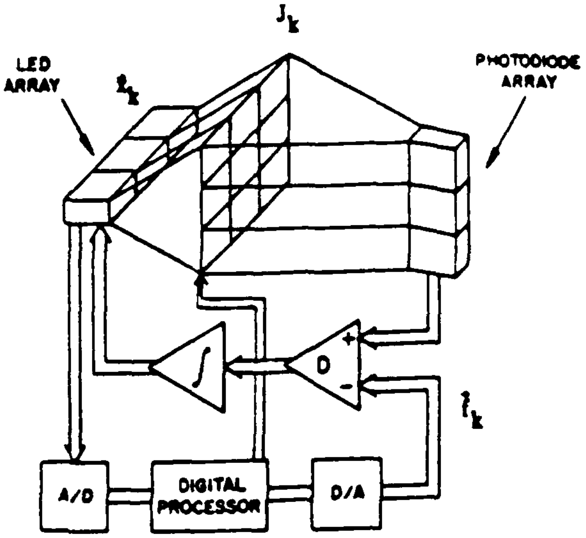


Fig.1 Block diagram of the hybrid optoelectronic system.

3.1 Hybrid Analog Optical Processor

In this system we use the optical analog processor to solve Eq. (8) approximately. For this system we introduce the following algorithm:

a) Use the digital processor to guess an initial solution $\vec{x}_0$.

b) Use the digital processor to compute both the vector $\vec{f}_k$ and the matrix J_k.

c) Use the optical analog processor to solve the system $J_k^o \vec{c}_k^o = \vec{f}_k^o$ for $\vec{c}_k^o$, approximately, where the superscript o's denote inaccuracies in optics or electronics.

d) Use the digital processor to read $\vec{c}_k^o$ and compute the refined solution $\vec{x}_{k+1} = \vec{x}_k - \vec{c}_k^o$.

e) Check if the norm $\|\vec{f}_{k+1} - \vec{f}_k\| < \epsilon$ stop, otherwise, go back to step (b) and recycle.

3.2 Hybrid BOC Processor

In this system the BOC is used to solve Eq.(8) exactly. For this system we introduce the following algorithm:

a) Use the digital processor to guess an initial solution $\vec{x}_0$.

b) Use the digital processor to compute both $\vec{f}_k$ and the matrix J_k.

c) Use the BOC to solve the system $J_k \vec{c}_k = \vec{f}_k$, exactly for $\vec{c}_k$.

d) Use the digital processor to read $\vec{c}_k$ and compute the refined solution $\vec{x}_{k+1} = \vec{x}_k - \vec{c}_k$.

e) Check if the norm $\|\vec{f}_{k+1} - \vec{f}_k\| < \epsilon$ stop, otherwise, go back to step (b) and recycle.

4. SPEED ANALYSIS

The following speed analysis is based on a system of linear equations with ·ze, n.

4.1 Digital Processor

The total time required, T_{DT}, to solve the system of nonlinear equations using a conventional digital processor is given by

$$T_{DT} = [\frac{n^3}{3} + 2n(n+1)]T_{D1}N_D \quad (9)$$

where

T_{D1} = the time needed to do one digital operation (e.g., a multiplication),

and

N_D = the number of iterations needed for the solution convergence.

4.2 Hybrid Analog Optical Processor

The total time required, T_{OA}, to solve the system of nonlinear equations using the processor introduced in Section 3.1 is given by

$$T_{OA} = [n(n+2)T_{D1} + T_{A1}]\, N_A \quad (10).$$

where

T_{A1} = the time required for the optical analog processor to solve the system of linear equations (8) approximately,

and

N_A = the number of iterations required for the solution convergence.

4.3 Hybrid BOC Processor

The total time required, T_{OB}, to solve the system of nonlinear equations using the processor introduced in Section 3.2 is given by

$$T_{OB} = [2n(n+1)T_{D1} + T_{A1}]\, I_B N_D \quad (11)$$

where

I_B = the number of iteration needed for the BOC to solve Eq. (8) to the specified accuracy.

4.4 Speed Advantage

It is of great interest to determine what is the break even point for the optical processor proposed to be faster than the digital processors. This condition is defined by

$$T_{DT} > T_{OA} \qquad (12)$$

and

$$T_{DT} > T_{OB} \ . \qquad (13)$$

From Eqs. (9) to (11) the conditions (12) and (13) can be written as

$$\left(\frac{n^2(n/3+1)}{I_A}\right) \times \left(\frac{T_{D1}}{T_{A1}}\right) > 1 \qquad (14)$$

or

$$A_n \times A_t > 1 \qquad (15)$$

for the hybrid analog processor, where

$$I_A = N_A/N_D . \qquad (16)$$

And for the hybrid BOC processor

$$\left(\frac{3^{-2n(n+1)(I_B-1)}}{I_B}\right) \times \left(\frac{T_{D1}}{T_{A1}}\right) > 1 \ , \qquad (17)$$

or

$$B_n \times A_t > 1 \ . \qquad (18)$$

Where

$$A_n = \frac{n^2(n/3+1)}{I_A} \ , \qquad (19)$$

$$B_n = \frac{n^3/3 - 2n(n+1)(I_B-1)}{I_B} \ , \qquad (20)$$

and

$$A_t = \frac{T_{D1}}{T_{A1}} . \qquad (21)$$

The number of iterations, I_A and I_B, usually are in the range of 1 to 10[1] The ratios, A_n and B_n are problem dependent, and are much larger than 1 for large values of n. On the other hand, A_t depends on the speed of the analog processor for solving a system of linear equation, which can be in the range of μsec. But since the matrix J_k need to be updated every cycle, writing the matrix J_k on the SLM becomes the bottleneck of the processor speed. With todays technology writing a matrix on an SLM may take a few milliseconds. So A_t is much less than 1. In Fig. 2(a) and (b) the Log (A_n) and Log (B_n) are plotted in terms of the system size, n, respectively. The ratio $A_n >$ 1, for $n \simeq 10$, while $A_n > 1$ for $n \cong 60$ and 120, for $I_B = 10$ and 20 respectively. For the A_t ratio in the range of 10^{-3}, we can have a speed advantage for the hybrid analog optical processor for $n \geq 50$, and for the hybrid BOC processor for $n \geq 120$.

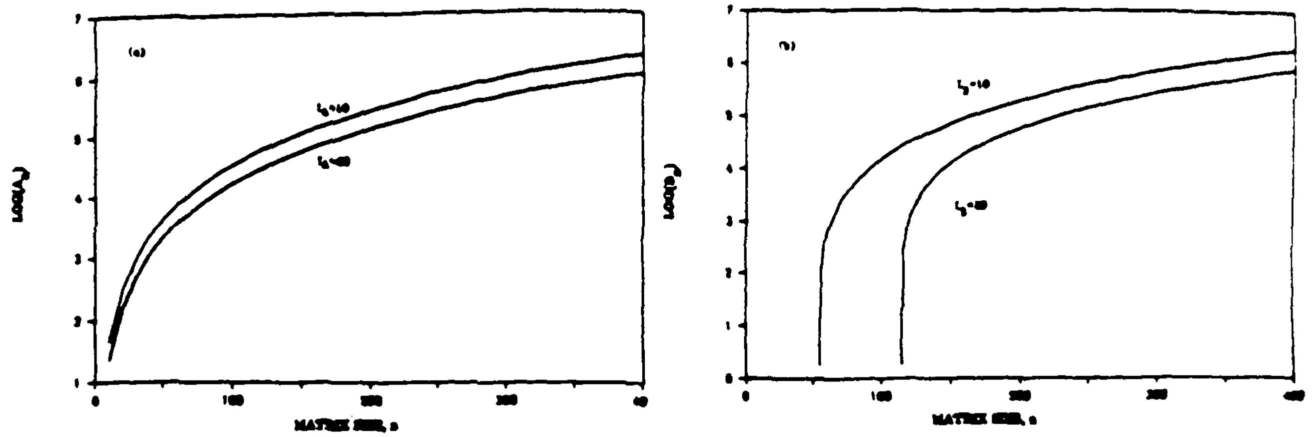


Fig.2 Plot of log of the ratio (a) A_n of Eq.(19), and (b) B_n of Eq.(20), in terms of the size of the matrix ,n.

Again this ratio A_t depends mainly on how fast we can write a matrix on the SLM. By the introduction of faster SLM's the speed advantage can be gained for smaller values of n.

5. CONCLUSIONS

Two new hybrid opto electronic processors are introduced for solving systems of nonlinear equations. The speed of the two processors is analyzed and compared with the speed of digital processors. It is shown that the main factor of the speed limitation is the speed the SLM's used to write the matrix on. With the existing SLM's a speed advantage can be gained for $n \geq 100$.

6. ACKNOWLEDGEMENTS

This research was supported by the Innovative Science and Technology Office of the Strategic Defense Initiative Organization, administered through the Office of Naval Research under contract N00014–86–k–0591, and by Rome Air Development Center, administered through Georgia Institute of Technology under contract E–21–669–S23.

REFERENCES

1. R. L. Johnson, Numerical Methods, John Wiley, New York (1982).
2. H. J. Caulfield, J. H. Gruniger, J. E. Ludman, K. Steiglitz, H. Rabitz, J. Gelfand, and E. Tsoni, "Bimodal Optical Computers," Appl. Opt. 25, 3123 (1986).
3. M. A. G. Abushagur and H. J. Caulfield, "Speed and Convergence of Bimodal Optical Computers," Opt. Eng. 26, 022 (1987).
4. M. A. G. Abushagur, H. J. Caulfield, P. M. Gibson and M. Habli, "Superconvergence of Hybrid Optoelectronic Processors," App. Oct. 26, 4906 (1987).

HYBRID ANALOG-DIGITAL LINEAR ALGEBRA PROCESSORS

H. J. Caulfield
Center for Applied Optics
The University of Alabama in Huntsville
Huntsville, AL 35899

Mustafa A.G. Abushagur
Department of Electrical and Computer Engineering
Center for Applied Optics
The University of Alabama in Huntsville

ABSTRACT

Analog optics is very fast but not very accurate. Digital electronics is much slower but much more accurate. Compromise (hybrid) systems appear to be intermediate in both speed and accuracy. As there are cases in which analog optics is too inaccurate <u>and</u> digital electronics is too slow; hybrid processors may have an important role to play.

I. ERRORS IN ANALOG OPTICS

Despite occasional claims in the literature, making a multichannel analog optical system with all channels controllable and repeatable to 1% absolute signal accuracy is extremely difficult. Thus if analog (number magnitude proportional to light irradiance) encoding is used, the accuracy with which numbers can be represented is, at best, 1% of the maximum magnitude number.

Unfortunately, 1% representation accuracy of inputs does not lead to 1% representation accuracy of calculated results. Obviously, the exact errors can not be predicted (otherwise they would hardly count as errors!). What we can predict is some sort of average or expected error.

Rather than predict errors in specific components of a vector or matrix, we seek more global metrics. The norm function $\| \cdot \|$ is convenient. The norm of the vector

$$\vec{V}^T = (V_1, V_2, \dots V_n)$$

is usually defined as

$$\| \vec{V} \|_N = [\, |V_1|^N + |V_2|^N + \dots |V_n|^N \,]^{1/N} \;.$$

Three N values are common.

$$N = 1 : \| \vec{V} \|_1 = |V_1| + |V_2| + \dots + |V_n| \;.$$

$$N = 2 : \| \vec{V} \|_2 = [\, |V_1|^2 + |V_2|^2 + \dots + |V_n|^2 \,]^{1/2} \;.$$

$$N \to \infty : \| \vec{V} \|_\infty = \max_{k=1,\dots,n} |V_n| \;.$$

Most mathematicians use the N = 2 (Euclidean) norm and drop the subscript, e.g.,

$$\| V \| = [\, V_1^2 + V_2^2 + \dots + V_n^2 \,]^{1/2}$$

We can now define a matrix norm

$$\| A \| = \max_{\vec{x}} \frac{\| A x \|}{\| x \|}$$

Since any $\vec{x}$ can be expanded in terms of the eigenvectors $\vec{e}_1, \vec{e}_2, \dots \vec{e}_n$ of A,

we have

$$\vec{x} = c_1 \vec{e}_1 + c_2 \vec{e}_2 + \dots + c_n \vec{e}_n \;.$$

But

$$A \vec{x} = c_1 \lambda_1 \vec{e}_1 + c_2 \lambda_2 \vec{e}_2 + \dots + c_n \lambda_n \vec{e}_n \;.$$

where

$$A \vec{e}_i = \lambda_i \vec{e}_i \quad .$$

Clearly

$$\| A \| = \max_i \, | \lambda_i | \quad .$$

Equally clear is the relationship

$$\| A^{-1} \| = \min_i \, (1/| \lambda_i |) .$$

A convenient measure of the ability of a matrix to lead to accurate results is the condition number. This is sometimes written as k(A), $\chi(A)$, or cond (A). For no particular reason, we choose $\chi(A)$. By definition,

$$\chi(A) = \| A \| / \| A^{-1} \| = | \lambda |_{max} / | \lambda |_{min} \quad .$$

Roughly speaking, the output error is $\chi(A)$ times the input or "representational" error in solving linear equations, inverting matrices, solving eigen problems, etc.

For these introductory purposes, these observations are sufficient. With representational accuracy fixed at 1% or less and requiring 1% or better accuracy in our results, we conclude that analog accuracy may suffice if $\chi(A) \leq 1$. Since, however

$$\chi(A) = | \lambda |_{max} / | \lambda |_{min} \; ,$$

we have $\chi(A) > 1$. In fact, we often have $\chi(A) \gg 1$. For these cases analog processors are hopelessly inaccurate. Accuracy is always lost.

II. PROBLEMS WITH DIGITAL OPTICS

Digital optics appears to offer a possible solution. Each number is represented by multiple analog channels in time and/or space. If multiplicity in space is used and proper number representation is employed, great parallelism and essentially-analog speed is accomplished at the price of great physical complexity. An additional problem in formatting and deformatting tends to slow the process and increase the power consumption. These problems may not be insurmountable but they are certainly difficult enough and far enough away from solution to motivate the search for alternative (non digital) ways of making accurate optical processors.

III. OBSERVATIONS ON COMPUTATIONAL COMPLEXITY

We want to have a tool for addressing the question: "How difficult is this calculation"? The now-traditional measure is computational complexity. The basic idea is to break up the operations into their most primitive parts, e.g., multiplies, and count the number of these required. Actually, we do one other important calculation. We associate a number N with the problem size, e.g., an N x N matrix has size N. We then ask how the number of calculations scales with N. Many algorithms, especially in linear algebra, have polynomial complexity. That is their complexity scales as roughly N^P, written $O(N^P)$ and often said "order of N^P." Note that it is the algorithm not the problem that has a complexity. Matrix-matrix multiplication as we all learned it is $O(N^3)$. Minimal complexity algorithms now approach $O(N^{2.5})$. This difference is far from subtle for large N.

Another way of viewing computational complexity is as a minimum price to be paid to make a calculation. That price can be paid in spatial complexity, temporal complexity or both. We will be aiming at high speed and thus low temporal complexity. To do this we will use a Bimodal Optical Computer (BOC) which does high complexity tasks by analog optics and lower (essentially by a factor of N) complexity task by digital electronics.

IV. ILLUSTRATIVE ALGORITHM

We suspect all linear algebra problems can be solved by BOC's. We will discuss the simple A x = b problem first. Other algorithms for other problems are shown in an appendix.

We suppose we have an analog optical Ax - b solver. We are given, to digital accuracy, A and b. We represent them in our optical solver as best we can. Our solution vector can be called x_0. To check whether x_0 is adequate or not we calculate (digitally, accepting x_0 as fully accurate) a residual

$$r_0 = b - Ax_0 \quad .$$

This is $O(N^2)$ digitally. If $| r_0 |$ is acceptably small, we quit. Otherwise we solve

$$A \, \Delta x_0 = r_0$$

optically. Note that

$$\begin{aligned} A \, (x_0 + \Delta x_0) &= Ax_0 + A\Delta x_0 \\ &= Ax_0 + r_0 \\ &= Ax_0 + (b - Ax_0) \\ &= b. \end{aligned}$$

Thus $x_0 + \Delta x_0$ is the desired x. Unfortunately, our analog solution for Δx_0 is inaccurate. Our result is not Δx_0 but δx_0.

We form

$$x_1 = x_0 + \delta x_0$$

digitally and evaluate

$$r_1 = b - Ax_1$$

digitally. If $| r_1 |$ is small enough, we quit. Otherwise we recycle.

That is the basic algorithm. It requires some modifications for use with optics. It also requires some convergence analysis. After all, if the analog solutions are inaccurate, isn't it possible that the solution will get worse not better?

V. CONVERGENCE

It is trivial to show that if we can guarantee $| r_k | < | r_{k-1} |$, then convergence must occur. In Ref. 1, we showed that this leads to the sufficient condition

$$\chi(A) \frac{\| E \|}{\| A \|} \quad \frac{1}{2} \ .$$

where E is the representation error matrix. If we like, we can rewrite this as

$$\| E \| < | \lambda |_{max} / 2\chi(A).$$

To first order it seems more profitable to assume that $\| E \| / \| A \|$ depends more on the computer than on the matrix A and can be replaced by a universal number ϵ, which we will call the computer accuracy. Then convergence occurs if

$$\chi \, (A) \, \epsilon < 1/2.$$

For $\epsilon = 0.01$, our hoped-for 1% accurate computer, we <u>strongly</u> <u>expect</u> convergence for

$$\chi \, (A) < 50.$$

We cannot guarantee convergence because it is the actual $\| E \| / \| A \|$ not its fictional problem-independent average that counts. Furthermore there is no reason to believe that convergence might not occur for much higher χ (A) values. We would expect that the probability of convergence is strongly related to

$$R = \chi(A) / \chi_{est}$$

where

$$\chi_{est} = 1/2 \, \epsilon$$

Thus we might expect convergence for virtually all R=1 problems and a much smaller fraction of R=100 problems. Even this statement hides a complexity. Given a problem and a computer, each particular incident (attempt to represent and solve the problem) leads to a different result. This can even be a strength if (a) we can afford the spatial or temporal complexity to calculate N independent answers and (b) we invoke the central limit theorem to suggest a roughly $\sqrt{N}$ improvement in ϵ.

VI. LORD KELVIN'S CONTRIBUTION

The basic approach of using a fast, low-accuracy processor in conjunction with a slow, high-accuracy computer is quite old. The history is available in Ref. 1 and references therein. Lord Kelvin (2) made a vital ribution: the proposal that the residual r be scaled to utilize the dynamic range of the processor well.

$$A \, \Delta x = r$$

ıld be multiplied by a scaler s to form

$$A \ (s\Delta x) = sr,$$

·e $\| sr \|_\infty = 1$ (the maximum representable number). It is not necessary to know s to high accuracy because ·ections to corrections are not first order critical. Since r is calculated digitally, we can also calcu- ; (to low accuracy)

$$s_\infty = 1/\| r \|_\infty$$

the same time.

VII. THE Ax = b SOLVER

We propose to use the O(1) time complexity, time continuous Ax = b solver of Cheng and Caulfield(3). The rt of the processor is the fully parallel Stanford matrix-vector multiplier. Figure 1 shows the system. ut lights representing x components are spread vertically across the columns attenuating mask representing Row sums of the transmitted light are detected to give components of the output vector y. For all k, we ow

$$\delta_k = b_k - y_k$$

ve x_k. Here y_k is a component of the calculated

$$y = Ax.$$

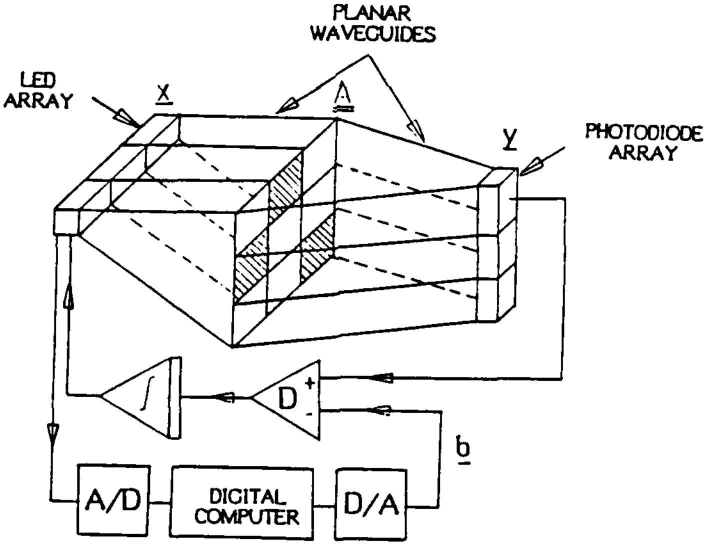


Fig. 1 System layout of the Bimodal Optical Computer (BOC)

Cheng and Caulfield showed that smooth convergence to the $| r | = 0$ solution occurs at a rate proportional o

$$e^{-\lambda_{min} \, t/t_0},$$

ʼhere λ_{min} is the eigenvalue of A with minimum $| \lambda |$ and t_0 is a characteristic time (roughly signal round trip ime). Obviously convergence requires

$$\lambda_{min} > 0.$$

\s it turns out (3), this is a sufficient condition for convergence. Calling the normalized time

$$\tau = t/t_0$$

and noting

$$\lambda_{min} = \lambda_{min}\ \lambda_{max}/\lambda_{max}$$

$$= \lambda_{max}/\chi(A),$$

we have a relaxation rate

$$e^{-\lambda_{max}\tau/\chi(A)}.$$

Thus many things affect the relaxation rate:

t_o,

the inherent processor speed,

λ_{max},

the maximum eigenvalue, and

$\chi(A)$,

the condition number of the problem

Thus even though the operation is temporally O(1), the convergence speed is clearly problem dependen Easy problems (low χ) converge rapidly. Hard problems, high χ, converge slowly.

VIII. SPEED ADVANTAGE

Let us compare a BOC Ax = b solver with a digital iterative Ax = b solver.

A single cycle requires one hybrid Ax = b solver cycle, T_H, plus temporally O(1) A/D and D/A operati plus 2N (N+1) digital electronic operations of duration T_{D1}. Taking the conversion times into T_H, we ha total time

$$T_H = I_H\ [T_{H1} + 2N\ (N+1)\ T_{D1}]$$

where I_H is the number of required iterations.

The iterative digital Ax = b solver requires a time

$$T_D = I_D\ [(N^3/3) + 2N^2\ T_{D1}].$$

We want

$$T_H << T_D$$

or

$$\underbrace{\left[\frac{(N^3/3) + 2N^2(1-k) - 2kN}{k}\right]}_{0\ (N^3)\ \text{PART}} \qquad \underset{\text{0(1) PART}}{\frac{T_{D1}}{T_{H1}}} >> 1.$$

Here

$$k = (I_H/I_D).$$

Let us consider the two factors separately.

The quantity

$$Q = \frac{T_{D1}}{T_{H1}}$$

problem dependent through T_H. For ultra fast electronics (nanoseconds) and small loops (t_o ~ 1 nosecond), the A/D and D/A converters may be the speed limiters. For 1% accuracy

$$e^{-\lambda_{min}t/t_o} \sim 0.01.$$

us

$$\frac{\lambda_{min}t}{t_o} \sim - \ell n\ (0.01)$$

$$T_H \sim - (t_o/\lambda_{min.})\ \ln\ (0.01)$$

ır $t_o \sim T_D$,

$$Q \sim t_o/T_H \sim + 4.6\lambda_{min}$$

.early

$$Q > 1,\ \text{for all}\ \lambda_{min} \geqslant,\ .22,$$

ıt perhaps not much less.

The $O(N^3)$, Ap, part depends strongly on k. Obviously k > 1. Hopefully we can keep k ~ 1. Let us examine ırious k > 0 case plotted in Fig. 2. We see that large advantages occur for low k (rapid convergence) and ırge N. That is we win for large, easy, or (preferably) large and easy problems. If, for example, Q = 0.1, e obtain a factor of 10 advantage for all k-N products above the horizontal line in Fig. 2. If the problem ize is, say, 200 and k ~ 1, the advantage can be many many orders of magnitude.

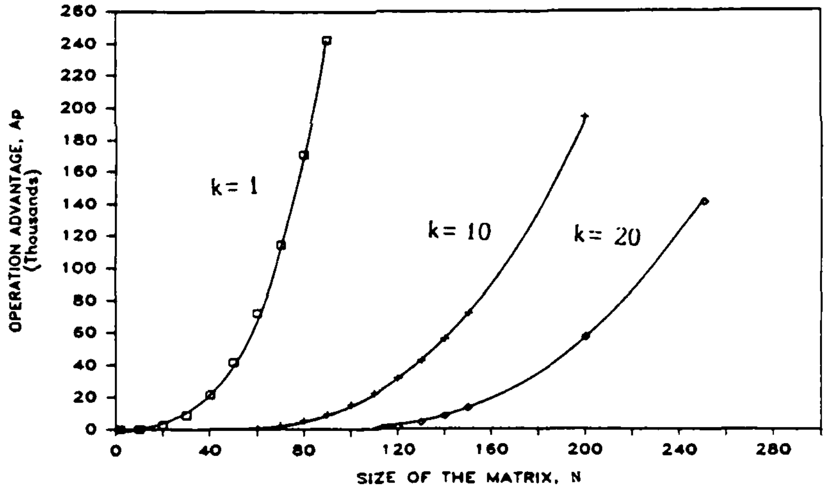


Fig. 2 The operation advantage, A_p

IX. SIMULATIONS

We can add Gaussian stochastic errors to the "true" numbers to simulatevarious accuracies. Figure 3 ıhows the I_H as a function of $\chi(A)$ for various problems, where $\| r \| / \| x \| < 10^{-6}$ is required. Note I_H / ː. since I_D may be 5 to 10. Thus 1:10^6 accuracy is a low k situation. Note as well, that convergence tends -o occur even for R > 1 (R = 4 in Fig. 3). We have achieved convergence for R's as high as 60. We might vant to relax from ϵ=0.01 (Fig. 3) to much less trying cases. Figure 4 shows that

$$I_H \propto \epsilon,$$

a very benign result.

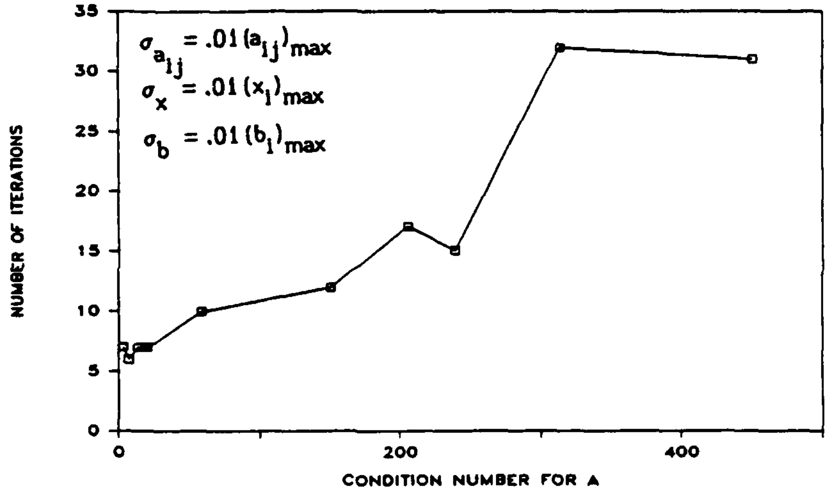


Fig. 3 Computer simulation results for the number of iterations needed for convergence of the solution, plotted vs. the condition number.

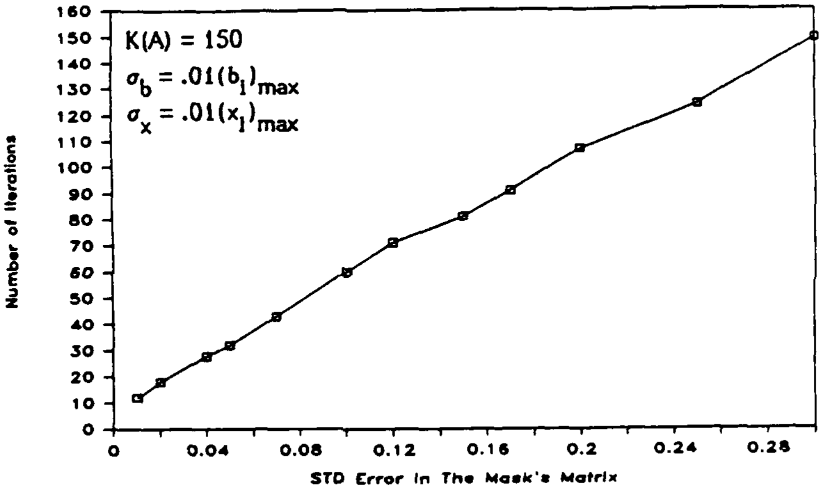


Fig. 4 The number of iterations plotted vs. the error's standard deviation in the matrix's mask, for a matrix with a condition number=150.

X. THE FUTURE

Besides building a moderately accurate BOC for testing, we will investigate improvements. Two imp ments are suggested below.

First, we can operate on A by "equilibration" to get an equivalent matrix A' such that

$$\chi(A') < \chi(A).$$

That is

$$\begin{aligned} Ax &= A\,(DD^{-1})x \\ &= (AD)\,(D^{-1}x) \\ &= A'\,x'. \end{aligned}$$

Here D is a diagonal matrix. For equilibration we might require row norm equality. In our early expe this led to roughly 20% improvement in condition number.

Second, we can use "convergence factors" to try to force or improve convergence.

rather than

$\vec{x}_k = \vec{x}_{k-1} - \Delta\vec{x}_{k-1}$.

se

$\vec{x}_f = \vec{x}_{k-1} - \theta\ {k-1}\ \Delta\vec{x}_{k-1}$.

·lever choice of the θ_k's we may be able to improve performance.

APPENDIX

In this appendix we present algorithms for solving the inverse of a matrix and the eigenvalue problems ıg BOC.

The Inverse of a Matrix

For an NxN matrix A the inverse matrix A^{-1} is defined to satisfy the following relationship

$AA^{-1} = I$, where I is the identity matrix (A1)
:h can be rewritten as

$$A[{-1 \atop i1}, A2{-1 \atop i2}, \ldots, A{-1 \atop in}] = I \qquad (A2)$$

re A^{-1} is the j-th column vector of the inverse matrix A^{-1} So eq. (A2) can written as n systems of lin- equations, which can be solved individually using the BOC as outlined in section IV.

Another method for solving the inverse matrix problem is by using the -1Pan-Reif method (4). If the

rix $B \approx A^{-1}$ then define the error matrix E as

$$E = I - BA \qquad (A3)$$

a^{-1} can be represented in terms b and E as

$$A^{-1} = (A-1B-1)B = (I-E)-1B \qquad (A4)$$

$$(1-x)^{-1} = \frac{1}{1-x} = 1+x+x^2+x^3+ \ldots.., \text{ for } x\ \ 1. \qquad (A5)$$

ıilarly,

$$A^{-1} = (1+ E + E^2 + E + E^3 + \ldots.)B \qquad (A6)$$

. if we start with an approximation for the inverse of A by B_0, then the error matrix E_1 will be given by (A7)

$$E_1 = I - B_0A$$

in more general form for an iteration k

$$E_k = I \qquad {}_{<-1}A \qquad (A7)$$

d

$$= (I + E_k + E_k^2 + \ldots..)\ B_{k-1} \qquad (A8)$$

n and Reif introduced a simple way of evaluating B_0 the initial approximation of A^{-1}. Define the factor t

$$t = \frac{1}{(\max\limits_{I} \sum\limits_{J} A(I,J) \mid (\max\limits_{I} \sum\limits_{J} \mid A(I,J) \mid)}, \qquad (A9)$$

ich is the product of the maximum magnitude of the sum of the rows of A by the maximum magnitude of the sum the columns of A. Now B_0 will be given by

$$B_0 = t\ A^H \qquad (A10)$$

ıere A^H is the Hermitian transpose of A

We now introduce an iterative method for solving the inverse problem. Eq. (A9) can be written as for the case of small error matrix

$$B_k \approx (I + E_k) B_{k-1} , k=1,2,... \quad \text{(A12)}$$

Now let us outline the iterative method for finding the inverse of A:

i) find $B_0 = t A^H$

ii) $E_k = I - B_{k-1}A$, $k=1,2,...$

iii) $B_k = (I + E_k) B_{k-1}$, $k=1,2,..$

iv) Cycle iteratively through steps ii) and iii) until all elements of E_k are within the required accuracy then terminate the proces and B_k is equal to the inverse of A. Doing this method using a digital computer requires a long time of operations and a large memory space for the matrix multiplication We can do this matrix multiplication using the BOC in mu less time with the same accuracies.

2. The Eigenvalue Problem

Determining the eigenvalues and their corresponding eigenvectors is a very fundamental and important problem in linear algebra. One of the most powerful method for determining the eigenvalues and eigenvectors is the inverse iteration method (5). For an NxN matrix A the eigenvalues λ_i and their corresponding eigenvectors $\vec{x}_i$ are defined by the equation

$$A\vec{x}_i = \lambda_i\vec{x}_i \quad \text{(A13)}$$

Let A has an n destinctive eigen values such that

$$|\lambda_1| > |\lambda_2| > ..... |\lambda_n| . \quad \text{(A14)}$$

Assume $q \approx \lambda_i$ then

$$(a = qI) \vec{y}^{(p+1)} = \vec{z}^{(p)} \quad \text{(A15)}$$

where

$$\vec{z}^{(p+1)} = \vec{y}^{(P+1)} / \| y^{(p+1)} \|_\infty \quad \text{(A16)}$$

$$\vec{y}^{(p)} = \vec{x}_1, \text{ and } 1/\| \vec{y}^{(p)} \|_\infty = \lambda_i - q \text{ as } p \to \infty \quad \text{(A17)}$$

so by assuming a value for the vector $\vec{z}^{(0)}$ then solving the system of equations given in Eq. (A15) we get $\vec{y}^{(1)}$ and from eq. (a16) compute $\vec{z}^{(1)}$ and we keep iterating until the vector y^P become stable then we terminate the iterations. This determines both the eigenvalue and the eigenvector very accurately.

The initial value of the eigenvalue q can be tetermined using Gerchgorin's theorem (6). for an nxn matrix A, let us define the radius $_k$ as

$$r_k = \sum_{\substack{j=1 \\ j=k}} | a_{kj} | \quad \text{(A18)}$$

where a_{kj} as the k,j coefficient of the matrix A. rk is a radius of disk Dk centered at a_{kk} within which an eigenvalde will lie

$$Dk = (\lambda | \; |\lambda - a_{kk}| \; r_k) \leq k = 1,2,3,....,N \quad \text{(A19)}$$

Then each eigenvalue of A must lie within the union s of these disks

$$S = \bigcup_{K=1}^{N} D_k \quad \text{(A20)}$$

So, after we determine the initial eigenvalues for the matrix A we can use the inverse iteration method to find the more refined values for the eigenvalues and their corresponding eigenvectors. In this process we will use the BOC to solve the set of equations (a15) within a reasonabie accuracy. this method will have the speed advantage over the all-digital processor because again we reduced the problem to a set of linear equation solution.

XIII. REFERENCES

1. H. John Caulfield, J. H. Gruniger, J. E. Ludman, K. Steiglitz, H. Rabitz and J. Gelfard, "Bimodal Optical Computers," Submitted to Applied Optics.
2. W. Thompson (Lord Kelvin), Proc. Roy. Soc. (London), vol. 28, p. 111 1878.
3. Wai K. Cheng and H. John Caulfield, "Fully Parallel Relaxation Algebraic Operations for Optical Computers," Opt. Comm., vol. 43, No. 4, p. 251 1982.
4. V. Pan and J. Reif, "Efficient Parallel Solution of Linear Systems," Proceedings of the 17th Annual ACM Symposium on Theory of Computing, May 1985.
5. J. H. Wilkingon, The Algebraic Eigenvalue Problem, Clarendon Press, Oxford (1965).
6. D. I. Steinberg, Computational Matrix Algebra, McGraw-Hill (1974).

JAM RESISTANCE OF THE BIMODAL OPTICAL COMPUTER

Mustafa A. G. Abushagur
Electrical and Computer Engineering Department
University of Alabama in Huntsville
Huntsville, Alabama 35899

H. John Caulfield
Center for Applied Optics
University of Alabama in Huntsville
Huntsville, Alabama 35899

ABSTRACT

The use of the Bimodal Optical Computer (BOC) in determining the weights for an adaptive phased array radar is introduced. Interference canceling is presented for two cases: first assuming the direction of the jammer is known, secondly no *a priori* information is assumed. Effect of the jammers on the array pattern is shown for up to four jammers.

1. INTRODUCTION

The sensitivity of a signal-receiving antenna array system to interfering noise sources can be reduced by suitable processing of the outputs of the individual array elements. The processing of the output of the array system acts as an adaptive filtering system [1-4]. The adaptive phased array radar systems provide the means of suppressing unwanted interference signals. This is achieved by nulling the array pattern at the direction of the jammers. Many algorithms have been introduced for the adaptation process and these are reviewed by Monzingo and Miller[2].

In this paper we present a new technique to determine the weights for the adaptive array using the bimodal optical computer (BOC)[5-7]. The bimodal optical computer is capable of solving systems of linear equation very rapidly with high accuracy. In the adaptation process we reduce the optimization problem to a system of linear equations, which in turn is solved using the BOC.

In Section 2 we review the basic theory of adaptive phased array radars. The bimodal optical computer algorithm for solving the optimization problem is presented in Section 3. Computer simulation results are given in Section 4. Conclusions and final remarks are given in Section 5.

2. ADAPTIVE PHASED ARRAYS

In adaptive phased array radars the incoming signal is detected by an array of sensors. The detected signal is a combination of the target signal plus interference and noise signals. The system is adjusted in such a way to suppress the interference signals reception without affecting the desired signal.

In this section we consider the two general cases of interference canceling: first by assuming that the interference signal direction is known; secondly by assuming no *a priori* information is known about the interference signal.

2.1 Interference Signal Direction is Known

When the interference signal direction is known the weights w_i's of the array can be chosen to suppress the interference signal. Let the system shown in Fig. 1 be used to demonstrate this adaptation technique.

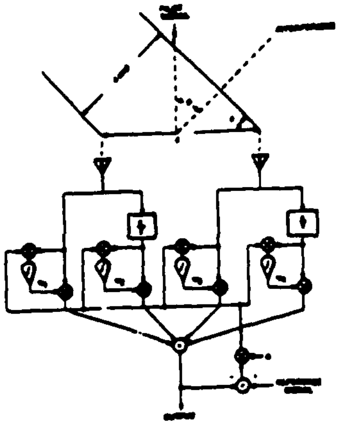


Fig. 1 Array configuration for interference canceling

The output signal of the array s(t) is given by [1]

$$
\begin{aligned}
s(t) = {} & P[(w_1 + w_3) \sin \omega_0 t + (w_2 + w_4) \sin(\omega_0 t - \theta - \tfrac{\pi}{2})] \\
& + I[w_1 \sin(\omega_0 t - \theta) + w_2 \sin(\omega_0 t - \theta - \tfrac{\pi}{2}) \\
& + w_3 \sin(\omega_0 t + \theta) + w_4 \sin(\omega_0 t + \theta - \tfrac{\pi}{2})] ,
\end{aligned} \tag{1}
$$

where

P = the pilot signal,

I = the interference signal, and

θ = the phase shift

$$\theta = \frac{2\pi d}{\lambda} \sin\phi. \tag{2}$$

To cancel the interference signal and to make the signal s(t) equal to the pilot signal, we need to solve the following system of linear equations for the weight w_i's:

$$
\left.
\begin{aligned}
& w_1 + w_3 = 1 \\
& w_2 + w_4 = 0 \\
& (w_1 + w_3) \, \mathrm{Cos}\theta - (w_2 - w_4) \sin\theta = 0 \\
& (w_2 + w_4) \, \mathrm{Cos}\theta + (w_1 - w_3) \sin\theta = 0 \; .
\end{aligned}
\right] \tag{3}
$$

The size of this system of linear equations depends on the number of sensors in the array. The number of jammers can make the system under or overdetermined, which are both time consuming algebra problems.

2.2 No *a priori* Information is Known

This is the most general case where we assume no information about jammers. The system used in this case is shown in Fig. 2.

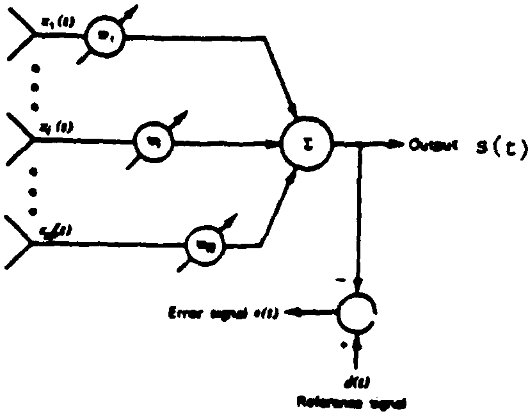


Fig. 2 Basic adaptive array structure with known desired signal

Each of the n sensors receives a signal $x_i(t)$ which is in turn multiplied by a variable weight w_i. The output signal s(t) is compared with the desired signal d(t), their difference, the error signal $\epsilon(t)$, is used to determine the value of w_i's. The output of the array is

$$s(t) = \sum_{i=1}^{n} x_i(t)\, w_i \tag{4}$$

or

$$s(t) = \vec{w}^T \vec{x} \tag{5}$$

where

$$\vec{w} = \begin{bmatrix} w_1 \\ \dot{w}_i \\ \dot{w}_n \end{bmatrix} \text{ and } \vec{x} = \begin{bmatrix} x_1(t) \\ \dot{x}_i(t) \\ \dot{x}_n(t) \end{bmatrix} . \tag{6}$$

For digital sampled data

$$s(j) = \vec{w}^T \vec{x}(j) \ , \tag{7}$$

and

$$\epsilon(j) = d(j) - s(j) = d(j) - \vec{w}^T \vec{x}(j) . \tag{8}$$

The optimum value of the weights, w_i's, is the one reduces $\epsilon(j)$ to zero or at least minimize it.

For N samples of data the optimum weights satisfy the following set of systems of linear equations:

$$\left.\begin{array}{l} \vec{w}^T \vec{x}(1) = d(1) \\ \cdots\cdots\cdots\cdots\cdots \\ \vec{w}^T \vec{x}(i) = d(i) \\ \cdots\cdots\cdots\cdots\cdots \\ w^T \vec{x}(N) = d(N) \; . \end{array}\right\} \qquad (9)$$

The N sets of equations have n unknowns, and usually N >> n, and are inconsistent and over specified. The optimization problem can be rewritten as

$$\vec{w}_{opt} = R_{xx}^{-1} \vec{r}_{xd} \qquad (10)$$

where

$$R_{xx} = E\{\vec{x}\vec{x}^T\} = E\left\{\begin{bmatrix} x_1x_1 & x_1x_n \\ x_2x_1 & x_2x_n \\ & \\ x_nx_1 & x_nx_n \end{bmatrix}\right\}, \qquad (11)$$

and

$$\vec{r}_{xd} = E\{\vec{x}d\}. \qquad (12)$$

The matrix R_{xx} is called the covariance matrix, where $E\{\cdot\}$ is the ensemble average.

Many algorithms are introduced[2] to solve for the weights in Eq. (10). Some of the popular algorithms are the least mean square (LMS), and the direct matrix inversion (DMI).

We'll briefly mention the DMI algorithm since it leads to the algorithm introduced in this paper. Eq. (10) cannot be determined exactly using a limited number of samples of the input data. For practical consideration a small number of samples is detected to be used in determining $\vec{w}$. The estimated value of Eq.(10) can be given by

$$\hat{\vec{w}} = \hat{R}_{xx}^{-1} \hat{\vec{r}}_{xd} \; . \qquad (13)$$

where

$\hat{R}_{xx}$ is the sample covariance matrix, and $\hat{\vec{r}}_{xd}$ is the sample cross-correlation vector, and are given by

$$\hat{R}_{xx} = \frac{1}{K} \sum_{j=1}^{K} \vec{x}(j)\, \vec{x}^T(j) \qquad (14)$$

and

$$\hat{r}_{xd} = \frac{1}{K}\sum_{j=1}^{K} \hat{x}(j)\, d(j) \ , \qquad (15)$$

and K is the number of samples. The DMI algorithm determines the inverse of the sample covariance matrix $\hat{R}_{xx}$, then from Eq. (13) evaluates $\hat{w}$.

3. THE BIMODAL OPTICAL COMPUTER ALGORITHM

Using either the LMS or the DMI algorithms, depends in its convergence on a number of factors, the most important one of them is the condition number of the matrix $\hat{R}_{xx}$. If the matrix $\hat{R}_{xx}$ is ill–conditioned or singular, it either converges very slowly or the inverse does not exist, respectively. In such cases other methods might be used, but they are lengthy and time consuming, so they are not suitable for a system where the time is a very crucial element.

We have shown in previous publications[6-7] that the bimodal optical computer is capable of solving such problems, where the system of equations is ill–conditioned, singular, overspecified or underspecified. The BOC is a hybrid system in nature, Fig. 3. It uses analog optics to solve the problem approximately but rapidly and it utilizes the digital electronics to refine the solution, in an interative scheme.

The optimization problem for the weights $\hat{w}$ introduced in Section 2, can be rewritten in the following form, from Eq. (13)

$$\hat{R}_{xx}\,\hat{w} = \hat{r}_{xd} \ . \qquad (16)$$

Eq. (16) is a system of linear equations can be solved using the bimodal optical computer. Among the advantages of using the BOC over the conventional techniques are: Speed, especially for large size arrays, convergence of the solution for difficult problems, ill–conditioned singular systems,which is the case of most of the adaptive array radar problems.

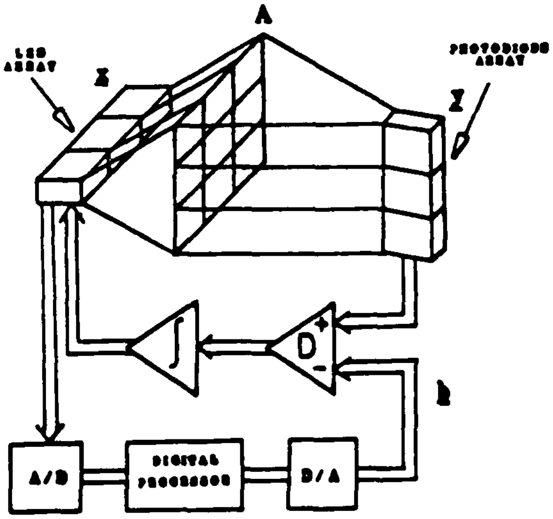


Fig. 3 The Bimodal Optical Computer

In the following section we present some of the preliminary results from computer simulation studies of the BOC in processing adaptive array problems.

4. SIMULATION RESULTS

Two simulation experiments are presented in this section. In the first experiment we used a five element array, and assumed the directions of the jammers are known. In the second experiment a 2 element array is used and no *a priori* information is assumed.

In Fig. 4 the array pattern of the 5 element array is plotted as a function of the angle, ψ. Fig. 4(a) shows the array pattern for the adaptation. In Fig 4(b) a jammer at 45o was considered, the pattern after adaptation is shown, the jammerknown. The array pattern after adaptation has reformed in such a way to null the jammer signal. In Fig. 4(c) four jammers are considered at 45°, 80°, 120° and 150°, the array patis again reformed to null all the jammers signals reception.

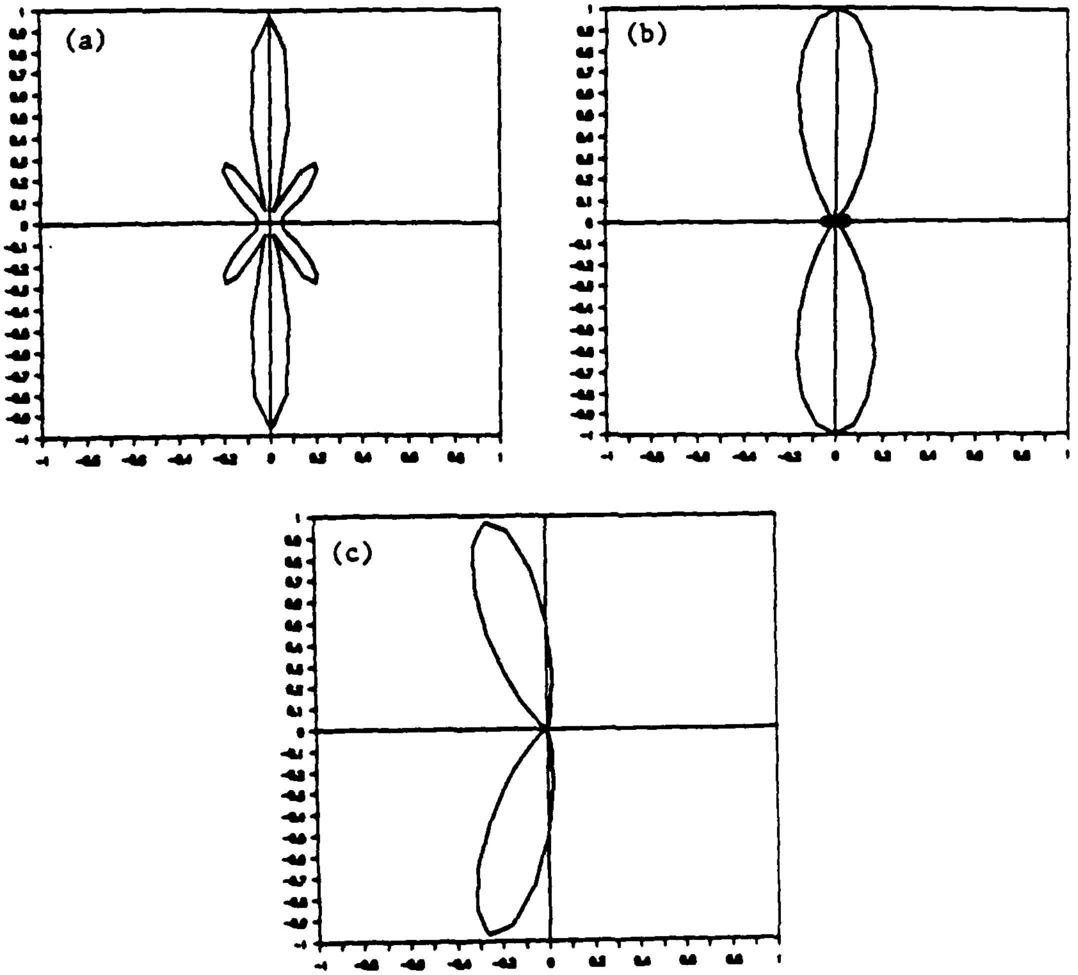


Fig. 4 Phased array pattern for 5 elements, (a) before adaptation, (6) adapted pattern for a jammer at 45o , and (c) adapted pattern for four jammers at 45o, 80o, 120o, and 150o.

In Fig. 5 the BOC was used to solve the adaptation problem assuming no a priori information about the interference signals. Fig. 5(a) shows the two-element array pattern before adaptation. In Fig. 5(b) to (d) the pattern is plotted for a single jammer placed at 30°, 45° and 60°, respectively. In all these plots the array adapted to cancel the interference signal in each of the given cases. In all of the above results the jammer signals is considered to be of the same strength as the desired signal, and the convergence of the solution obtained in less than five iterations. Also the condition number of the R_{xx} is between 10^6 and ∞.

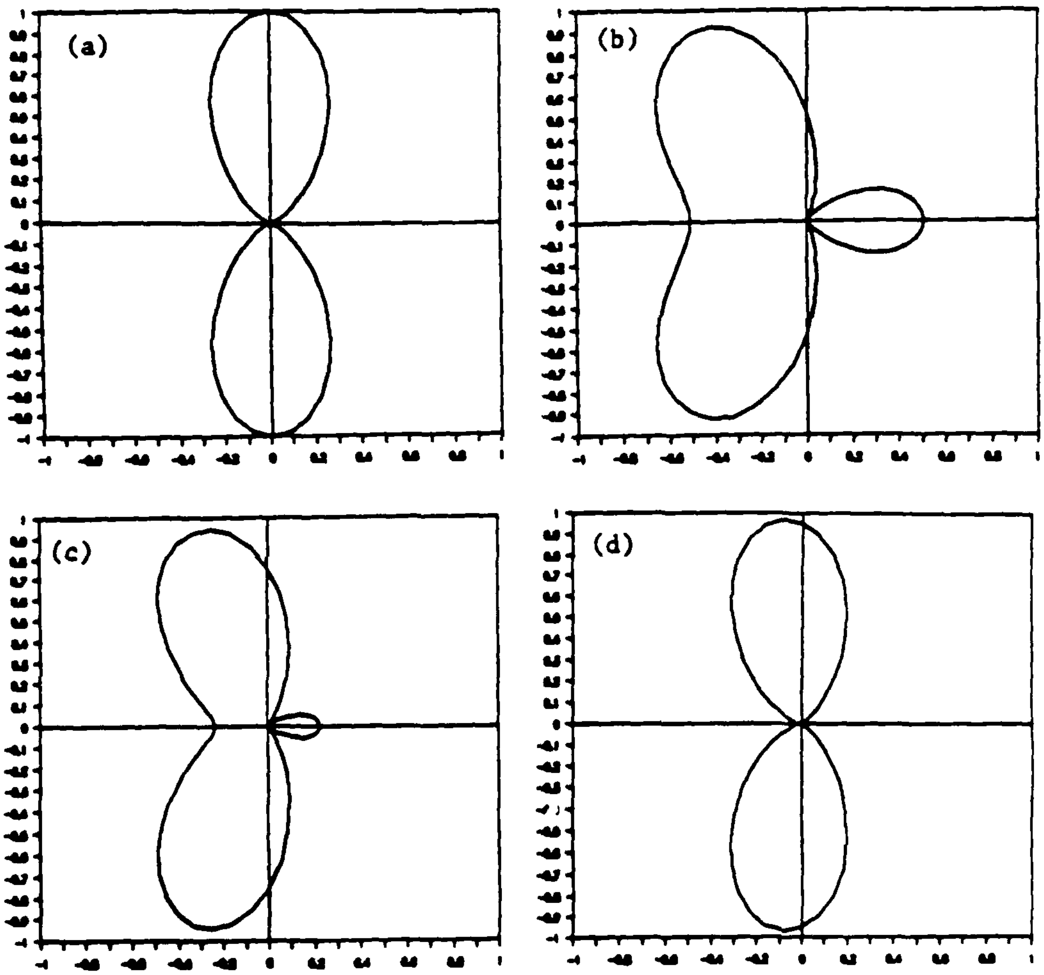


Fig. 5 Two-element phased array pattern, (a) before adaptation, (b) to (d) adapted patterns for single jammers at 30o, 45o and 60o, respectively.

4. CONCLUSIONS

The bimodal optical computer is shown in these preliminary results to present a powerful mean in solving adaptive phased array problems. We are considering in the future work larger array sizes, receiver noise, and very strong interference signals.

5. ACKNOWLEDGEMENTS

This research was supported by the Innovative Science and Technology Office of the Strategic Defense Initiative Organization, administered through the Office of Naval Research under contract N00014-86-k-0591.

REFERENCES

1. B. Widrow, P. E. Mantey, L. J. Griffiths and B. B. Goode, "Adaptive Antenna Systems," Proc. IEEE, 55, p. 2143 (1967).

2. Robert M. Monzingo and Thomas W. Miller, Introduction to Adaptive Arrays, John Wiley and Sons, New York (1980).

3. Bernard Widrow and Samuel D. Stearns, Adaptive Signal Processing, Prentice-Hall, Englewood Cliffs, New Jersey (1985).

4. S. Haykin, Editor, Array Signal Processing, Prentice-Hall, Englewood Cliffs, New Jersey (1985).

5. H. J. Caulfield, J. H. Gruniger, J. E. Ludman, K. Steiglitz, H. Rabitz, J. Gelfand, and E. Tsoni, "Bimodal Optical Computers," Appl. Opt. 25, 3123 (1986).

6. M. A. G. Abushagur and H. J. Caulfield, "Speed and Convergence of Bimodal Optical Computers," Opt. Eng. 26, 022 (1987).

7. M. A. G. Abushagur, H. J. Caulfield, P. M. Gibson and M. Habli, "Superconvergence of Hybrid Optoelectronic Processors," App. Oct. 26, 4906 (1987).

ADAPTIVE ARRAY RADAR DATA PROCESSING USING THE BIMODAL OPTICAL COMPUTER

Mustafa A. G. Abushagur
Electrical and Computer Engineering Department
University of Alabama in Huntsville
Huntsville, Alabama 35899

KEY TERMS

Adaptive arrays, optical computing, optical data processing

ABSTRACT

The use of the bimodal optical computer (BOC) in determining the weights for an adaptive phased array radar is introduced. Interference canceling is presented for two cases: (1) assuming the direction of the jammer is known and (2) assuming no a priori information. The effect of the jammers on the array pattern is shown for up to four jammers.

1. INTRODUCTION

The sensitivity of a signal-receiving antenna array system to interfering noise sources can be reduced by suitable processing of the outputs of the individual array elements. The processing of the output of the array system acts as an adaptive filtering system [1–4]. The adaptive phase array radar systems provide the means of suppressing unwanted interference signals. This is achieved by nulling the array pattern in the direction of the jammers. Many algorithms have been introduced for the adaptation process and they are reviewed by Monzingo and Miller [2].

In this paper we present a new technique for determining the weights for the adaptive array using the bimodal optical computer (BOC) [5–7]. The bimodal optical computer is capable of solving systems of linear equation very_rapidly with high accuracy. In the adaptation process we reduce the problem to a system of linear equations, which in turn is solved using the BOC.

In Section 2 we review the basic theory of adaptive phased array radars. The bimodal optical computer algorithm for solving the adaptation problem is presented in Section 3. Computer simulation results are given in Section 4. Conclusions and final remarks are given in Section 5.

2. ADAPTIVE PHASED ARRAYS

In adaptive phased array radars the incoming signal is detected by an array of sensors. The detected signal is a combination of the target signal plus interference and noise signals. The system is adjusted in such a way to suppress the interference signal reception without affecting the desired signal.

In this section we consider the two general cases of interference canceling: (1) by assuming that the interference signal direction is known and (2) by assuming no a priori information is known about the interference signal.

2.1. Interference Signal Direction is Known. When the interference signal direction is known the weights w_i of the array can be chosen to suppress the interference signal. Let the system shown in Figure 1(a) be used to demonstrate this adaptation technique. The output signal of the arrray $s(t)$ is given by [1]

$$\begin{aligned} s(t) = P\big[(w_1 + w_3)\sin\omega_0 t + (w_2 + w_4)\sin(\omega_0 t - \theta - \tfrac{1}{2}\pi)\big] \\ + I\big[w_1\sin(\omega_0 t - \theta) + w_2\sin(\omega_0 t - \theta - \tfrac{1}{2}\pi) \\ + w_3\sin(\omega_0 t + \theta) + w_4\sin(\omega_0 t + \theta - \tfrac{1}{2}\pi)\big]. \end{aligned} \quad (1)$$

where

$$\begin{aligned} P &= \text{the pilot signal}, \\ I &= \text{the interference signal}, \\ \theta &= \text{the phase shift}, \\ \theta &= \frac{2\pi d}{\lambda}\sin\psi. \end{aligned} \quad (2)$$

To cancel the interference signal and to make the signal $s(t)$ equal to the pilot signal, we need to solve the following system of linear equations for the weights w_i:

$$\begin{aligned} w_1 + w_3 &= 1, \\ w_2 + w_4 &= 0, \\ (w_1 + w_3)\cos\theta - (w_2 - w_4)\sin\theta &= 0, \\ (w_2 + w_4)\cos\theta + (w_1 - w_3)\sin\theta &= 0. \end{aligned} \quad (3)$$

The size of this system of linear equations depends on the number of sensors in the array. The number of jammers can make the system under or overdetermined, both of which are time consuming algebra problems.

2.2. No A Priori Information is Known. This is the most general case where we assume no information about jammers. The system used in this case is shown in Figure 1(b). Each of the n sensors receives a signal $x_i(t)$ that is in turn multiplied by a variable weight w_i. The output signal $s(t)$ is compared with the desired signal $d(t)$ and their difference, the error signal $\epsilon(t)$, is used to determine the value of w_i. The output of the array is

$$s(t) = \sum_{i=1}^{n} x_i(t) w_i \quad (4)$$

or

$$s(t) = \mathbf{w}^T\mathbf{x}, \quad (5)$$

where

$$\mathbf{w} = \begin{bmatrix} w_1 \\ \dot{w}_i \\ \dot{w}_n \end{bmatrix} \quad \text{and} \quad \mathbf{x} = \begin{bmatrix} x_1(t) \\ \dot{x}_i(t) \\ \dot{x}_n(t) \end{bmatrix}. \quad (6)$$

For digital sampled data

$$s(j) = \mathbf{w}^T\mathbf{x}(j) \quad (7)$$

and

$$\epsilon(j) = d(j) - s(j) = d(j) - \mathbf{w}^T\mathbf{x}(j). \quad (8)$$

The optimum value of the weights w_i is the one that reduces $\epsilon(j)$ to zero or at least minimizes it.

For N samples of data the optimum weights satisfy the following set of systems of linear equations:

$$\begin{aligned} \mathbf{w}^T\mathbf{x}(1) &= d(1) \\ &\vdots \\ \mathbf{w}^T\mathbf{x}(i) &= d(i) \\ \mathbf{w}^T\mathbf{x}(N) &= d(N). \end{aligned} \quad (9)$$

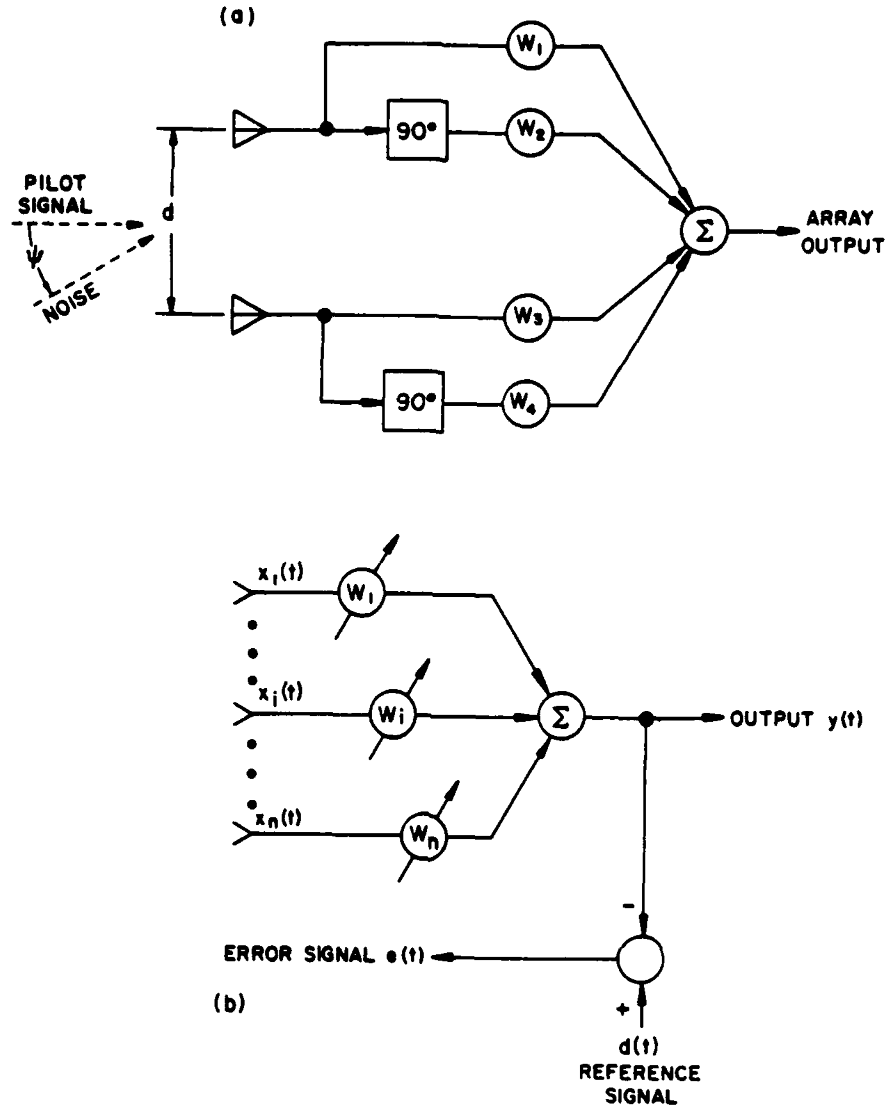


Figure 1 Basic adaptive array system with (a) signal and noise directions known and (b) no a priori information assumed.

The N sets of equations have n unknowns, and usually $N \gg n$, and are inconsistent and overspecified. The optimization problem can be rewritten as

$$\mathbf{w}_{\mathrm{opt}} = R_{xx}^{-1}\mathbf{r}_{xd}, \tag{10}$$

where

$$R_{xx} = E\{\mathbf{x}\mathbf{x}^T\} \tag{11}$$

and

$$\mathbf{r}_{xd} = E\{\mathbf{x}d\}. \tag{12}$$

The matrix R_{xx} is called the covariance matrix, where $E\{\cdot\}$ is the ensemble average.

Many algorithms are introduced [2] to solve for the weights in Eq. (10). Some of the popular algorithms are the least mean square (LMS) and the direct matrix inversion (DMI).

We will briefly mention the DMI algorithm since it leads to the algorithm introduced in this paper. Equation (10) cannot be determined exactly using a limited number of samples of the input data. For practical consideration a small number of samples is detected to be used in determining $\mathbf{w}$. The estimated value of Eq. (10) can be given by

$$\hat{\mathbf{w}} = \hat{R}_{xx}^{-1}\hat{\mathbf{r}}_{xd}, \tag{13}$$

where $\hat{R}_{xx}$ is the sample covariance matrix and $\mathbf{r}_{xd}$ is the sample cross-correlation vector that are given by

$$\hat{R}_{xx} = \frac{1}{K}\sum_{j=1}^{K}\mathbf{x}(j)\mathbf{x}^T(j) \tag{14}$$

and

$$\hat{\mathbf{r}}_{xd} = \frac{1}{K}\sum_{j=1}^{K}\mathbf{x}(j)\,d(j); \tag{15}$$

K is the number of samples. The DMI algorithm determines the inverse of the sample covariance matrix $\mathbf{R}_{xx}$ and then from Eq. (13) evaluates $\hat{\mathbf{w}}$.

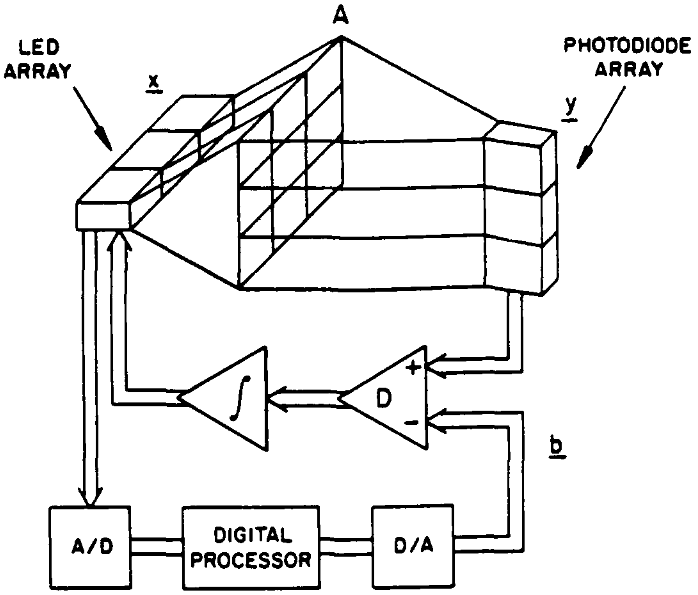


Figure 2 The bimodal optical computer used in solving a system of linear equations.

3. THE BIMODAL OPTICAL COMPUTER ALGORITHM

Convergence of either the LMS or the DMI algorithms depends on a number of factors, the most important being the condition number of the matrix $\hat{R}_{xx}$. If the matrix $\hat{R}_{xx}$ is ill-conditioned or singular, it either converges very slowly or the inverse does not exist, respectively. In such cases other methods might be used, but they are lengthly and time consuming, so they are not suitable for a system where time is a very crucial element.

We have shown in previous publications [6, 7] that the bimodal optical computer is capable of solving such problems, where the system of equations is ill-conditioned, singular, overspecified, or underspecified. The BOC is a hybrid system by nature; see Figure 2. It uses analog optics to solve the problem approximately but rapidly and it utilizes the digital electronics to refine the solution, in an iterative scheme.

The adaptation problem for the weights $\mathbf{w}$ introduced in Section 2, can be rewritten in the following form, from Eq. (13):

$$\hat{R}_{xx}\hat{\mathbf{w}} = \hat{\mathbf{r}}_{xd}, \qquad (16)$$

which can be written as

$$A\mathbf{x} = \mathbf{b}, \qquad (17)$$

where

$$A = \hat{R}_{xx},$$
$$\mathbf{x} = \hat{\mathbf{w}},$$
$$\mathbf{b} = \mathbf{f}_{xd}.$$

Equation (16) is a system of linear equations that can be solved using the bimodal optical computer. Among the advantages of using the BOC over the conventional techniques are speed, (especially for large size arrays), convergence of the solution for difficult problems, and ill-conditioned singular systems, which is the case for most of the adaptive array radar problems.

We review here the BOC algorithm in solving the system $A\mathbf{x} = \mathbf{b}$.

(a) Solve $A\mathbf{x} = \mathbf{b}$ using the analog optical processor to get $\mathbf{x}_0$.

(b) With a dedicated digital electronics processor, read $\mathbf{x}_0$ and evaluate the residue

$$\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0 = A\mathbf{x} - A\mathbf{x}_0 = A(\Delta\mathbf{x}_0). \quad (18)$$

(c) Normalize $\mathbf{r}_0$ to use the dynamic range of the system.

(d) Solve optically the system

$$A\mathbf{z} = s\mathbf{r}_0 \qquad (19)$$

where

$$\mathbf{z} = s(\Delta\mathbf{x}_0), \qquad (20)$$

and s is the radix used in normalizing $\mathbf{r}$.

e) Evaluate electronically

$$\mathbf{x}_1 = \mathbf{x}_0 + \Delta\mathbf{x}_0, \qquad (21)$$

and

$$hr_1 = \mathbf{b} - A\mathbf{x}_1. \qquad (22)$$

(f) If $|\mathbf{r}_1|$ is small enough, stop. Otherwise, go to (c) and recycle.

In the following section we present some of the preliminary results from computer simulation studies of the BOC in processing adaptive array problems.

4. SIMULATION RESULTS

Two simulation experiments are presented in this section. In the first experiment we used a five element array and assume

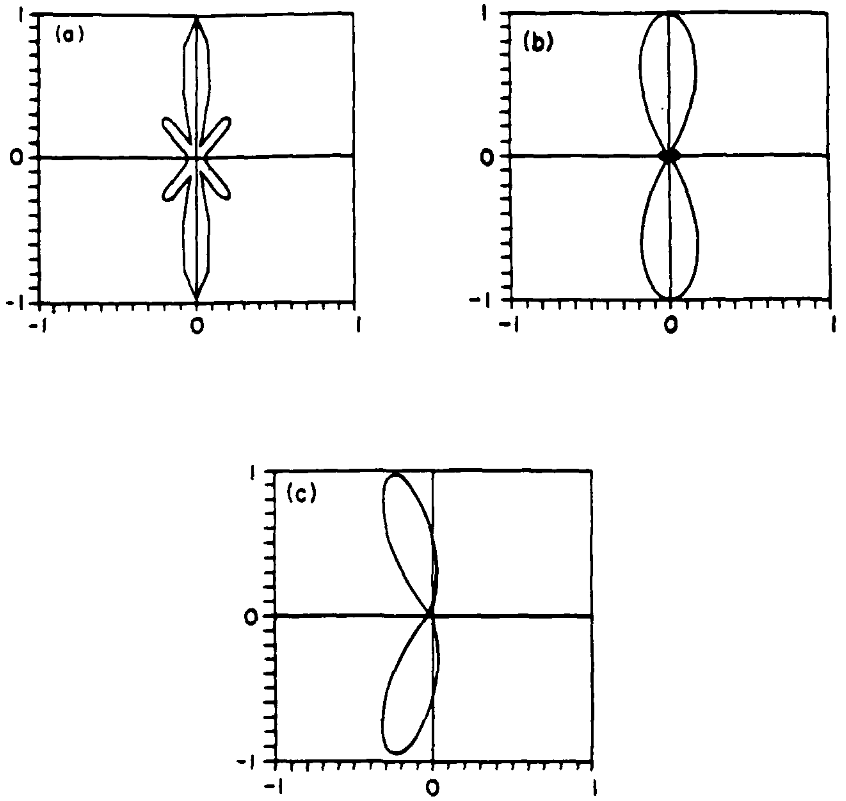


Figure 3 Phased array pattern for five elements (a) before adaptation, (b) adapted for a jammer at 45°, and (c) adapted for four jammers at 45°, 80°, 120°, and 150°.

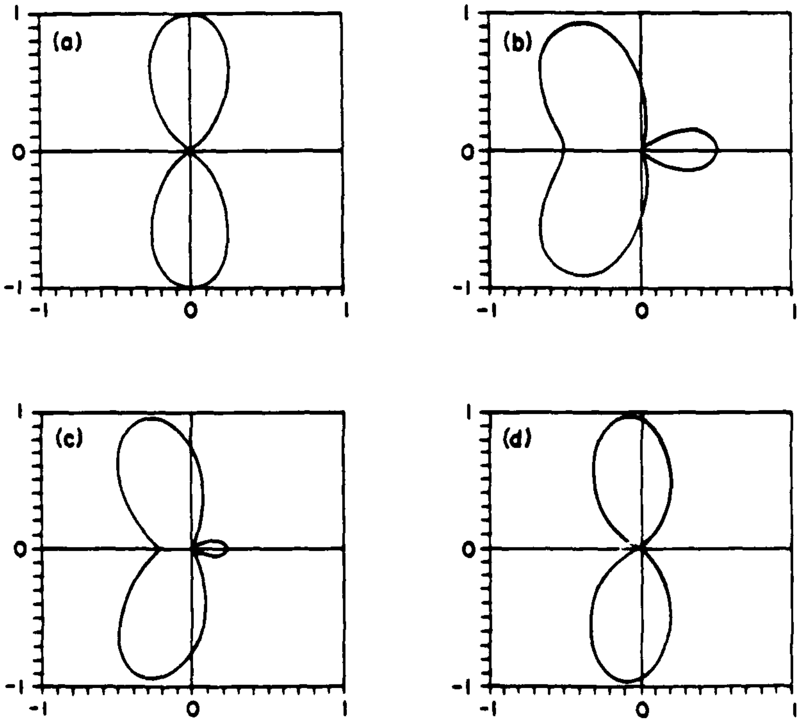


Figure 4 Two element phased array pattern (a) before adaptation and (b)–(d) adapted for single jammers at 30°, 45°, and 60°, respectively.

the directions of the jammers were known. In the second experiment a two element array is used and no a priori information is assumed.

In Figure 3 the five element array pattern is plotted as a function of the angle ψ. Figure 3(a) shows the array pattern before adaptation. In Figure 3(b) the pattern after adaptation is shown for a jammer at 45°. The array pattern after adaptation has reformed in such a way that it nulls the jammer signal. In Figure 3(c) four jammers are considered at 45°, 80°, 120°, and 150°. The array pattern is again reformed to null all the jammers signal reception.

In Figure 4 the BOC was used to solve the adaptation problem assuming no a priori information about the interference signals. Figure 4(a) shows the two-element array pattern before adaptation. In Figure 4(b)–(d) the pattern is plotted for a single jammer placed at 30°, 45°, and 60°, respectively. In all these plots the array adapted to cancel the interference signal in each of the given cases. In all of the preceding results the jammer signals are considered to be of the same strength as the desired signal, and the convergence of the solution obtained in less than five iterations. Also the condition numbered of the R_{xx}' is between 10^6 and ∞.

5. CONCLUSIONS

The bimodal optical computer is shown in these preliminary results to present a powerful mean for solving adaptive phased array problems. We are considering in future work larger array sizes, receiver noise, and very strong interference signals.

ACKNOWLEDGMENTS

The author would like to thank Dr. H. John Caulfield for numerous enlightening discussions.

This research was supported by the Innovative Science and Technology Office of the Strategic Defense Initiative Organization, administered through the Office of Naval Research under contract N00014-86-k-0591, and by Rome Air Development Center, administered through Georgia Institute of Technology under contract E-21-669-S23.

REFERENCES

1. B. Widrow, P. E. Mantey, L. J. Griffiths, and B. B. Goode, "Adaptive Antenna Systems," *Proc. IEEE*, vol. 55, 1967, p. 2143.
2. R. M. Monzingo and T. W. Miller, *Introduction to Adaptive Arrays*, Wiley, New York, 1980.
3. B. Widrow and S. D. Stearns, *Adaptive Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1985.
4. S. Haykin ed., *Array Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1985.
5. H. J. Caulfield, J. H. Gruniger, J. E. Ludman, K. Steiglitz, H. Rabitz, J. Gelfand, and E. Tsoni, "Bimodal Optical Computers," *Appl. Opt.*, Vol. 25, 1986, p. 3123.
6. M. A. G. Abushagur and H. J. Caulfield, "Speed and Convergence of Bimodal Optical Computers," *Opt. Eng.*, Vol. 26, 1987, p. 22.
7. M. A. G. Abushagur, H. J. Caulfield, P. M. Gibson, and M. Habli, "Superconvergence of Hybrid Optoelectronic Processors," *Appl. Opt.*, Vol. 26, 1987, p. 4906.

Received 6-20-88

Microwave and Optical Technology Letters, 1/7, 236–240

CCC 0895-2477/88/$4.00

Highly precise optical-hybrid matrix processor

Mustafa A.G. Abushagur

Electrical and Computer Engineering Department and Center for Applied Optics
University of Alabama In Huntsville, Huntsville, Alabama 35899

H. John Caulfield

Center for Applied Optics
University of Alabama In Huntsville, Huntsville, Alabama 35899

Abstract

An optical-hybrid matrix processor is presented and compared in its speed with a digital electronic processor. Optical-hybrid matrix processors are shown to be far more superior in their speed in solving systems of linear equations. This advantage in speed increases with the increase of the matrix size. The problem of the convergence of the solution using the optical-hybrid is investigated. It is found that even with using elctro-optical systems with an error as high as 5% in the I/O devices, convergence was achieved for matrices with condition numbers as high as 150. Some means of improving the condition number of a matrix are also intoduced.

I. Introduction

Analog optics is very attractive for signal processing and computing because of its ability to process two-dimenstional data in parallel very rapidly. Unfortunately, this high speed parallel processing achieves only low accuracy because of the nature of the analog processing especially in the optical systems. These accuracy problems rise from errors,[1] in representing and reading the signal using the electro-optic I/O devices. The method introduced by Caulfield[1] (which is outlined in section II of this paper) combines the high speed and parallelism of the optical processor and the high accuracy of the digital computer, using Lord Kelvin's iterative method.[2] In section II of this paper we present a comparison between the time required to solve a system of linear equations using the optical-hybrid processor to that required by the digital processor. In section III we present a numerical analysis of the convergence of the solutions for a linear algebraic equations as a function of the condition number of the matrix and the errors in representing the I/O data in the optical system, using computer simulation of the optical-hybrid processor. In section IV a conclusion and final remarks are drawn.

II. Computation speed analysis

The optical-hybrid processor works in the following manner for a system linear equations (it is also applicable to other problems- both linear and nonlinear),

$$A\,\underline{x} = \underline{b}\ , \tag{1}$$

where A is an nxn matrix, $\underline{x}$ and $\underline{b}$ are nx1 vectors.
a) Using an optical analog processor we can calculate an approximate solution $\underline{x}^o$ of the linear system, the superscript o's indicate inaccuracies in the optics and electronics, so the equations solved by the optical processor are

$$A^o\,\underline{x}^o = \underline{b}^o\ . \tag{2}$$

b) Remember the solution to a high accuracy with the digital computer. Use a dedicated digital processor to calculate the residue

$$\underline{r} = \underline{b} - A\,\underline{x}^o = A\,(\,\underline{x} - \underline{x}^o\,) = A\Delta\,\underline{x}\ . \tag{3}$$

c) Use the optical analog processor to solve the linear equations

$$A^o\,\underline{y} = s\,\underline{r}^o\ ,\quad \text{where } \underline{y} = s\,\Delta\underline{x}, \tag{4}$$

for $\underline{x}$, where s is a "radix", or scale factor chosen to make a good use of the dynamic range.
d) Use the digital processor to refine the solution for $\underline{x}^1$.

$$\underline{x}^1 = \underline{x}^o + \Delta\,\underline{x}\ . \tag{5}$$

If the refined solution x^1 is accurate enough terminate the iterations. Otherwise go back again to d), c) and d) for a more refined solution following the above outlined procedure.

To get some quantitative values for the speed of this process compared to that carried by the digital computer, we will calculate the number of operations required by each method then multiply it by the time required by each operation, we are going to consider the number of operations regardless if they are additions or multiplications.

Let us consider an nxn matrix A , the time required for one iteration of the procedure outlined above, T_{O1}, is given by

$$T_{O1} = \quad T_{A1} + 2n\ (\ n + 1\ )\ T_{D1}\ , \tag{6}$$

where T_{A1} = the time required to solve $A\ \underline{x} = \underline{b}$ by analog optics,
and T_{D1} = the time required to make one digital operation.
Therefore the time required to make I_O iterations with the optical processor is given by

$$T_O = I_O\ \{\quad T_{A1} + 2n\ (\ n + 1\ )\ T_{D1}\}\ . \tag{7}$$

While the time required by the digital computer to solve the linear equations using the Cholesky's method[3] in I_D iterations takes the time, T_D, given by

$$T_D = (\ n^3\ /\ 3 + 2\ n^2\ )\ I_D\ T_{D1} \tag{8}$$

The condition which we need to satisfy to have an advantage in time in using the optical processor over the digital processor is

$$T_O << T_D\ . \tag{9}$$

Therefore, for a clear time advantage for the optical-hybrid processor, from Eqs. (7)-(9) we want that

$$I_O\{\quad T_{A1} + 2\ n\ (\ n + 1\ )\ T_{D1}\} << (\ n^3\ /\ 3 + 2\ n^2\ )\ I_D\ T_{D1} \tag{10}$$

or

$$k\ \{\quad T_{A1} + 2\ n\ (\ n + 1\ )\ T_{D1}\} << (\ n^3\ /\ 3 + 2\ n^2\ )\ T_{D1} \tag{11}$$

where $k = I_O\ /\ I_D$. Eq. (11) can be rewritten in the following form

$$\frac{n^3\ /\ 3 + 2\ n^2\ (\ 1 - k\ ) - 2\ k\ n}{k} \qquad T_{D1}\ /\ T_{A1} \quad >> \quad 1\ . \tag{12}$$

The advantage of using the optical-hybrid processor over the digital processor in speed is obvious from Eq. (12), and it increases by the increase of the size of the matrix n. To examine this condition very carefully, let us rewrite Eq. (12) in the following form

$$A_p\ A_I \quad >> 1\ , \tag{13}$$

where

$$A_p = 2\ [\ n^3/6 + n^2\ (\ 1 - k\ ) - n\ k\ ]\ /\ k\ , \quad \text{and} \tag{14}$$

$$A_I = \ T_{D1}\ /\ T_{A1}\ . \tag{15}$$

Here A_I is an "inherent advantage". A single analog operation is much faster than the digital one. The whole $A\underline{x}=\underline{b}$ solution will be slower than a single digital operation, but the analog optical $A\underline{x} = \underline{b}$ solver works at speeds independent of n. On the otherhand, T_{D1} is operation dependent, also it includes the time in performing the operation and in storing and retrieving the data from the memory of the computer,which is a time consuming especially with the increase of n.

A_p is a problem related advantage, it a function of the size of the matrix n and the ratio of iterations k. The operation advantage A_p is ploted in Fig. 1 as a function of n and k. It is clear that A_p increases very rapidly by the increase of the size n, even if the number of iteration in the optical-hybrid processing scheme are much larger than those for the digital processing, while in reality they will be approximately the same for the same problem conditions.

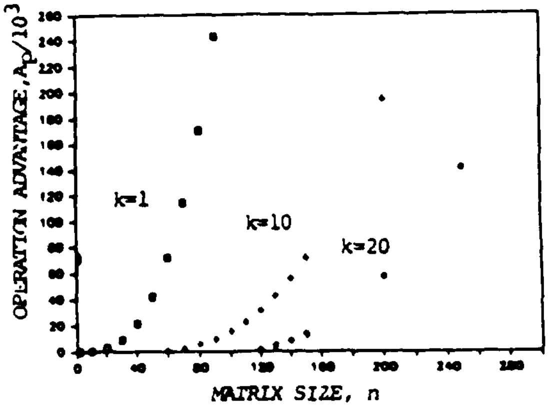


Fig.1 The operation advantage, A_p in Eq.(4) plotted vs. n in terms of k=1,10 and 20.

III. Convergence of the solution

The block diagram of the optical-hybrid processor is shown in Fig. 2. The solution of the linear algebraic equation will be done optically using the method introduced by Cheng and Caulfield[4]. The question of the convergence in discussed in Cheng and Caulfield paper and it is found that if the matrix has positive eigenvalues than the solution will converge regardless of the size of the matrix. This simply applies to step c) OF THE PROCEDURE OUTLINED IN SECTION II. We turn next to the total process.

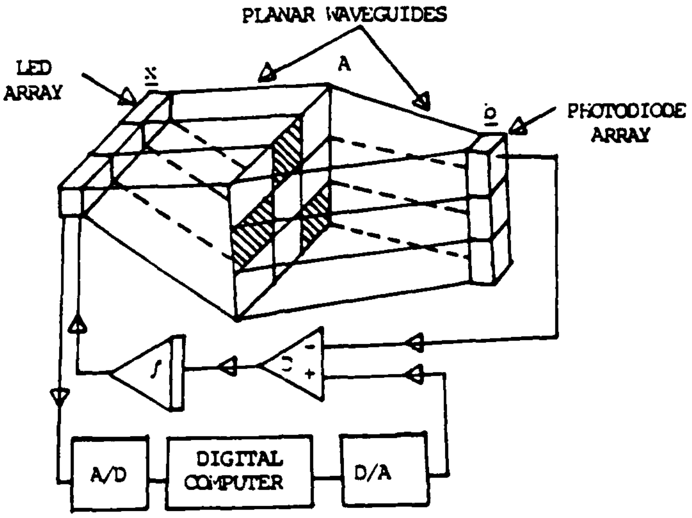


Fig.2 System layout of the optical-hybrid processor.

In this section of the paper we present a numerical analysis of the convergence of the solution and its dependence on the condition number of the matrix. The condition number of the matrix A is defined as

$$K (A) = ||A|| \cdot ||A^{-1}|| \qquad (16)$$

where $||\cdot||$ is the norm of the matrix. The condition number is a measure of the accuracy of the A $\underline{x}$ = $\underline{b}$ solutions. The larger the condition number the less accurate the result achieved with any fixed accuracy computer. In this paper we report a simulation of the system shown in Fig. 2 by a computer algorithm to study the convergence of the solution of the linear equation. The computer algorithm simulates the analog optical processor and the electro-optic I/O devices in such a way that allows us to control the errors occuring in representing the matrix by an optical mask, and also the error in reading the photodiode voltage and in converting the input in the system to light by the LED's. To simulate the experimental enviroment we have used a Gaussian random number generator to generate the error signals.

The curve shown in Fig.3 is the result of a simulation experiment for the optical-hybrid processor with the following charactaristics: The matrix A can be represented by an optical mask (a photographic film or a spatial light modulator) with an error of stadard deviation of 1% of the maximum coefficient of the matrix. The vector $\underline{x}$ can be real with an error of standard deviation = 1% of the maximum element of the vector $\underline{x}$, also the error standard deviation in representing $\underline{b}$ by the photodiode is 1%. From Fig.2 we see that the solutions converge with an error less than one millionth (or any other accuracy) even for condition number 500. For condition numbers less than 250 the number of iterations required are less than 20. In order to guarantee convergence with 1% accuracies, we must restrict matrices to condition numbers less than 50.

To study the effect of the error in representing the matrix by an optical mask on the number of iterations to get the solution within 10^{-6} accuracy, we have changed the standard deviation of the error in representing the matrix over the range from 1% to 30% for a condition number 150 and we calculated the number of iterations required for each case. The relatioship between the number of iterations and the standard deviation of the error in representing the matrix is plotted in Fig.4. As the error increases the number of iterations increase in an almost linear way. Even for an error of 30% in representing the matrix, the solution still converges. This intresting result proves that even by using inaccurate optics,optical-hybrid processor can still solve the linear system of equations very accurately

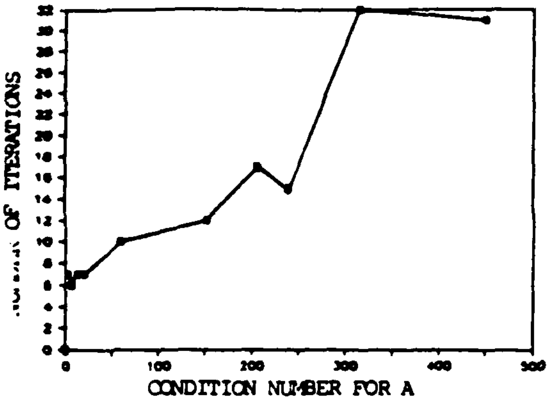


Fig.3 Computer simulation results for the number of iterations needed for convergence of the solution plotted vs. the condition number of A.

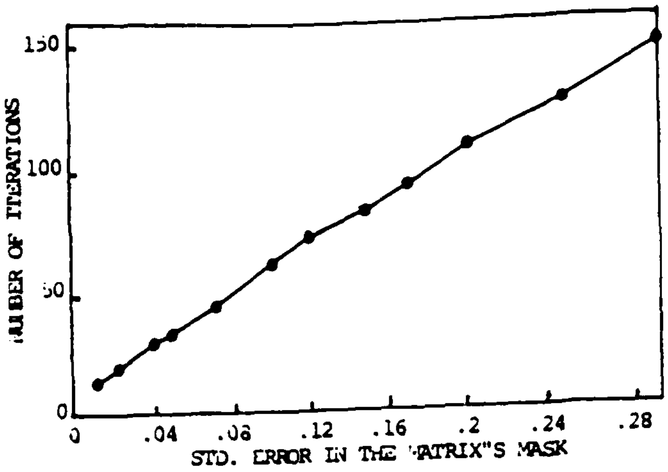


Fig.4 The number of iterations plotted as a function of the error's standard deviation in the matrix's mask, for a matrix with a condition number=150.

The condition number is one of the determining factors of the speed of convergence of the solutions as can be seen in Fig.4. Smaller condition numbers yield faster convergence of the solution. In searching for a way to improve the condition number of a given matrix, we found one way of doing that is by normalizing the matrix in the following manner

$$a_{i1} = a_{i1} \ / \ (\ a_{i1}^2 + a_{i2}^2 + \ldots\ldots\ldots\ldots + a_{in}^2 \ )^{1/2} \quad ; \quad i = 1,2,\ldots\ldots\ldots, n \tag{17}$$

wher a_{ij}'s are the coefficients of the matrix A. This normalization decreases the value of the condition number of the matrix which in turn increases the speed of the convergence process. Fig.5 shows a plot of the condition number before and after the normalization of the matrix, from which we can see an improvement in the condition number after normalization.

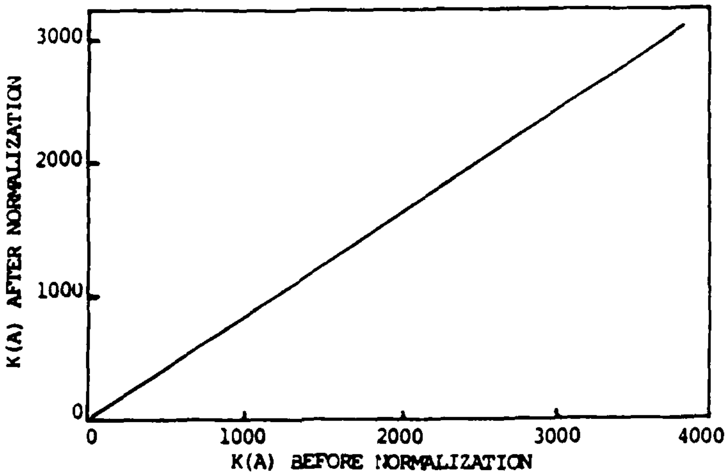


Fig.5 The condition number of the matrix after it has been normalized is plotted versus the original condition number.

IV. Conclusions

The optical-hybrid matrix processor discussed in this paper have shown very promising results, it is clearly very compatable in both speed and accuracy with the digital processor, in solving a system of linear equations. The advantage of the speed of the processor increases with the increase of the size of the matrix. The analysis carried out in this paper is not limited to the solution of a system of linear equations but is applicable as well to other linear and nonlinear problems. Another interesting result presented here is that the optics which is used in the processor can have a tolerence of 5 to 10% without sacrificing the accuracy of the solution, although it is shown that the less error in both optics and electronics the faster the solution will converge.

Acknowledgments

This research was supported by the Innovative Science and Technology Office of the strategic Defence Initiative Organization and administrated through the Office of Naval Research under contract N00014-85-k-6479.

References

1. H. John Caulfield, J.H. Gruniger, J.E. Ludman, K. Steiglitz, H. Rabitz and J. Gelfand, "Bimodal Optical Computers," submitted to Applied Optics.
2. W. Thompson (Lord Kelvin), Proc. Roy. Soc. (London), Vol. 28, p. 111 1878.
3. G.W. Stewart, Introduction to Matrix Computatios, Academic Press 1973.
4. Wai K. Cheng and H. John Caulfield, "Fully parallel Relaxation Algebraic Operations for Optical Computers," Opt. Comm., Vol 43, No.4, p.251 1982.

SOLVING EIGENVALUE PROBLEMS USING THE BIMODAL OPTICAL COMPUTER

Mustafa A.G. Abushagur
Department of Electrical and Computer Engineering
and Center for Applied Optics
University of Alabama in Huntsville

H. John Caulfield
Center for Applied Optics
University of Alabama in Huntsville
Huntsville, AL 35899

ABSTRACT

An algorithm for computing the eigenvalues and the corresponding eigenvectors of a matrix using the bimodal optical computer (BOC) is presented. Accuracy of the solutions are similar to that of the digital computer. The speed of the computation is compared to the existing super ccmp'iters. The BOC is shown to have advantage in speed especially for large size matrices. The advantage in speed increases by the increase of the size of the matrix.

I INTRODUCTION

Eigenvalue problems arise in many physical problems. The eigenvalue solutions ar3 often performed by iterarive methods using digital computers.[1] Solving this class of problems is a time consuming. The time required for determining the eigenvalues and eigenvectors increases with the size of the matrix. For matrices of very high rank the digital computer bocome very slow. Optics appears to be a natural candidate for tackling such a class of problems. Previous work on optical eigenvalue processors offers potential accuracy problems.[2,3,4]

In this paper we introduce a method to determine the eigenvalues and their corresponding eigenvectors for a positive definite matrix using the bimodal optical computer (BOC).[5,6] The accuracy of the solution is equivalent to that of a floating point processors because of the hybrid nature of the BOC provided that convergence occurs at all. The method is outlined in section II. The speed of the algorithm is analyzed in section III. Conclusions are drawn in Section IV.

II EIGENVALUE ALGORITHM

For an nxn matrix A the eigenvalues and the eigenvectors are given by

$$A \vec{e}_i = \lambda_i \vec{e}_i \quad ,i=1,2,.....,n \qquad (1)$$

where λ_i's are the eigenvalues of the matrix A and $\vec{e}_i$'s are the corresponding eigenvectors. In this paper we are going to consider the case where the eigenvalues of the matrix are all positive, real and not equal, i.e.

$$\lambda_1 \rangle \lambda_2 \rangle \lambda_3 \rangle \ldots\ldots \rangle \lambda_n \rangle 0 \ . \tag{2}$$

There are many methods for determining the eigenvalue and eigenvectors of a matrix. One of the powerful methods is the inverse iteration method. [4,7] The inverse iteration method is outlined as follows:

a) Assume an initial value for the eigenvalue $\lambda_i = q_i$ and an eigenvector $\vec{z}^{(0)}$. The assumption for the initial value of the eigenvalue can be done using the *Gershgorin* circle theorem.

b) Then solve the system of linear equations

$$(A-q_i I)\ \vec{y}^{(p+1)} = \vec{z}^{(p)}\ ,\ p=0,1,2,3,\ldots.. \tag{3}$$

where $\vec{z}^{(p+1)} = \vec{y}^{(p+1)} / \|\vec{y}^{(p+1)}\|_\infty$, (4)

$\|\cdot\|_\infty$ is the infinite norm, [8] and I is the identity matrix.

As $p\to\infty$ $\vec{y}^{(p)}=\vec{e}_i$ and $1/\ \|\vec{y}^{(p)}\|_\infty = \lambda_i - q_i$. (5)

Of course other norms will work and even work somewhat better, but the infinite norm is very easy to calculate.

The time consuming operation in this method is solving the system of linear equations in (3) for $\vec{y}^{(p+1)}$. This system of equations can be solved using the bimodal optical computer (BOC) very rapidly relative to electronics, especially for large n.[5] The algorithm which we propose in this paper for determining the eigenvalues and the eigenvectors using the BOC is as follows:

a) Assume a value for q_i and $\vec{z}^{(0)}$ using the digital processor.

b) Solve the linear system of equations

$$(A-q_iI)\ \vec{y}^{(1)} = \vec{z}^{(0)} \qquad (6)$$

for $\vec{y}^{(1)}$ using the BOC.

c) Compute the *norm* $||\vec{y}^{(1)}||_\infty$ and $\vec{z}^{(1)}$ using a dedicated digital processor.

d) If $||\vec{y}^{(1)}||_\infty - ||\vec{y}^{(0)}||_\infty \le \epsilon$, where ϵ is the error acceptable in computing the eigenvalues and the eigenvectors, then stop the iterations otherwise go back to step b).

In this algorithm we use the analog optics to compute an approximate solution for the system of linear equations which is then refined using the digital processor. This refined solution has the digital computer accuracy but determined much faster. This computation is done using the BOC which is shown in Fig.1. The convergence of the solution of the system of linear equations using the BOC is discussed in the paper of Abushagur and Caulfield.[6]

III Speed of the Algorithm

In this section we present a comparison between the speed of the digital computer in determining one eigenvalue for the matrix A to that of the bimodal optical computer.

The time required for doing one iteration of the procedure outlined in Sec.II using the digital computer is[8]

$$T_D = (7n^3/4)T_{D1}, \tag{7}$$

where T_{D1} is the time required for one digital operation. The time required to do one iteration, T_O, using the BOC is given by

$$T_O = [T_{A1} + 2n(n+2)\ T_{D1}]\ I_O, \tag{8}$$

where T_{A1} is the time required to solve the system of linear equations by the analog processor and I_O is the number of iteration required in refining the solution of the system of linear equations using the BOC. For a clear advantage in speed for the BOC over the digital computer we need to satisfy the following condition.

$$T_D \gg T_O, \tag{9}$$

or,

$$([\ 7n^3/4 - 2n(n+2)\ I_0]/I_0)\ (T_{D1}/T_{A1})\ \gg 1. \qquad (10)$$

Eq.(10) can be rewritten as

$$A_p \cdot A_I \gg 1. \qquad (11)$$

where $A_I = T_{D1}/T_{A1}$ and $A_P = [7n^3/4 - 2n(n+2) I_0]/I_0$. (12)

T_{D1} and T_{A1} are independent on the size of the matrix. For a rough comparison

$$T_{A1} \simeq 2\ \mu sec, \qquad (13)$$

and

$$T_{D1} \simeq 1\ \mu sec, \text{ for a typical microcomputer and,} \qquad (14)$$

$$\simeq 1 \text{ nsec, for a CRAY2.} \qquad (15)$$

If we substitute from Eqs.(13) and (15) into Eq.(11) the condition for the advantage in speed for the BOC over the CRAY2 will be

$$A_p \gg 2000\ . \qquad (16)$$

In Fig.2 $A_p A_I$ is plotted as a function of the size of the matrix n usin T_{D1} of the CRAY2 computer. It is clear that the BOC can have an advantage of speed over the CRAY2 if the size of the matrix is in the range of 50 or larger. This advantage in speed increases by the increase of the size of the matrix. Which makes this method very attractive for such a class of problems.

IV CONCLUSION

A new method for solving the eigenvalue problem using the bimodal optical computer is presented. It is shown that for a well conditioned matrix the solution for the eigenvalues and eigenvectors can be achieved much more faster using the BOC than the existing supercomputers. This advantage in speed becomes very clear for large size matrices.

AKNOWLEDGEMENT

This research was supported by the Innovative Science and Technology Office of the Strategic Defence Initiative Organization and administrated through the office of Naval Research under contract N00014-86-K-0591.

References

1. A.R. Gourlay and G.A Watson,Computational Methods for Matrix Eigenproblems (John Wiley 1973).

2. H. John Caulfield,D. Dvore,J.W. Goodman,and William Rhodes,"Eigenvector Determination by noncoherent optical methods",App. Opt. 20, pp. 2263-65 (1981).

3. H.J. Caulfield and J. Gruninger,"Algorithm Improvements for Optical Eigenfunction Computers,"App. Opt. 22, p. 2075 (1983).

4. H.J. Caulfield and J. Gruninger,"Relaxation Method for Eigenvector Solutions," Opt. Comm., 46, p. 83 (1983).

5. H.J. Caulfield,J.H. Gruninger, J.E. Ludman, K. Steiglitz, H. Rabitz and E. Tsoni,"Bimodal Optical Computers," App. Opt. 25, pp. 3128-31 (1986)

6. Mustafa A.G. Abushagur and H. John Caulfield,"Speed and Convergence of Bimodal Optical Computers," to be published in the January 1987 issue of Optical Engineering.

7. J.H. Wilkinson,The Algebraic Eigenvalue Problem (Clarendon, Oxford 1965).

8. R.L. Johnson,Numerical Methods(John Wiley,New York 1982).

FIGURE CAPTIONS

Fig.1 Block diagram for the bimodal optical computer (BOC) for solving the system of linear equations $A\vec{x}=\vec{b}$.

Fig.2 The speed advantage A_p A_I for the BOC over the CRAY2 in solving the eigenvalue problem.

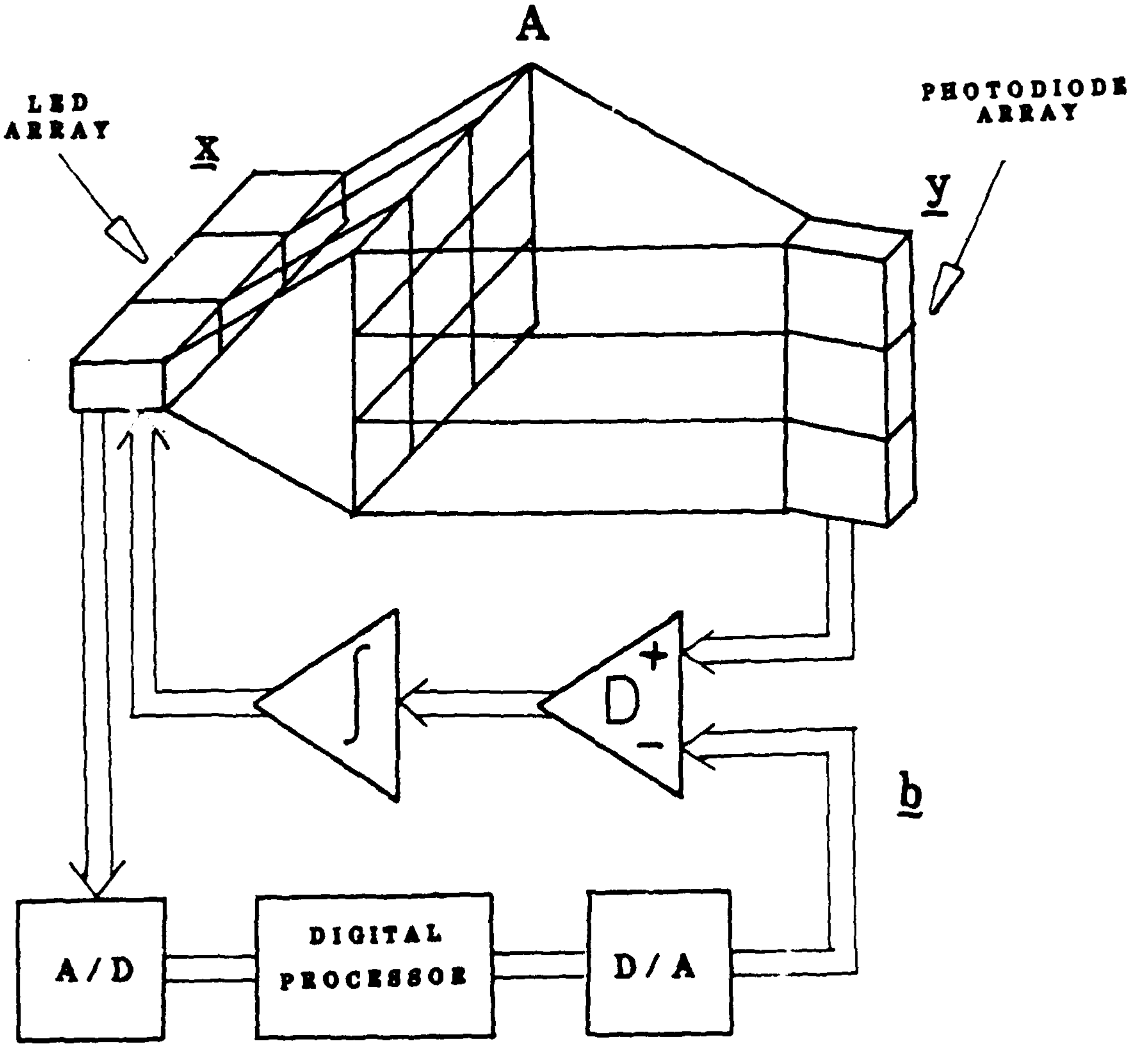


Fig.1 Block diagram for the bimodal optical computer (BOC) for solving the system of linear equations $A\vec{x}=\vec{b}$.

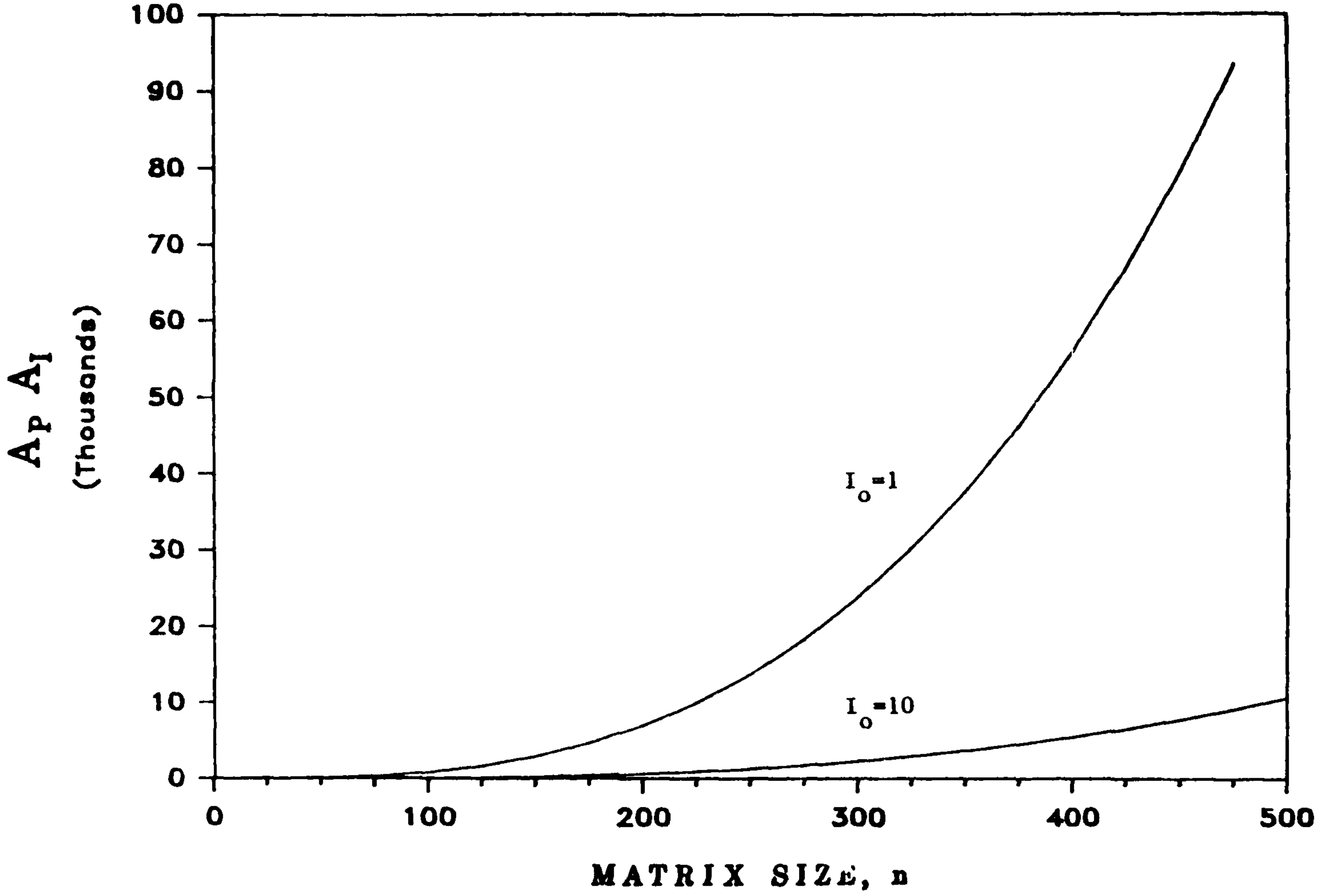


Fig.2 The speed advantage $A_P A_I$ for the BOC over the CRAY2 in solving the eigenvalue problem.

SOLVING SYSTEM OF LINEAR EQUATIONS USING THE BIMODAL OPTICAL COMPUTER (EXPERIMENTAL RESULTS)

M. A. Habli
M. A. G. Abushagur
Electrical and Computer Engineering Department
University of Alabama in Huntsville
Huntsville, Alabama 35899

H. J. Caulfield
Center for Applied Optics
University of Alabama in Huntsville
Huntsville, Alabama 35899

ABSTRACT

Hardware and software design of the Bimodal Optical Computer (BOC) and its implementations are presented. Experimental results of the BOC for solving a system of linear equations $A\underline{x} = \underline{b}$ is reported. The effect of calibration, the convergence reliability of the BOC, and the convergence of problems with singular matrices are studied.

1.INTRODUCTION

Analog optical systems are becoming very attractive in the area of signal processing because of their ability to process in parallel two dimensional data very rapidly. However, analog optical systems have low accuracy. BOC [1–4] solves this low accuracy problem, by using a combination of both analog optical system and digital processor.

In this paper we present experimental results using BOC for solving systems of linear equations. In Section 2 a comparison between astigmatic optics and waveguides based algebra processors is presented. The hardware and the software design of BOC is in Section 3. Section 4 contains the experimental results of the BOC for solving a system of linear equations. The conclusions are in Section 5.

2.ASTIGMATIC OPTICS AND WAVEGUIDES BASED ALGEBRA PROCESSORS

The analog optical system can be applied in many applications. This paper concentrates on solving a system of linear equations. Goodman [5] has introduced an astigmatic processor to perform matrix vector multiplications, which can also be used in a system of linear equations solver. However, the main problem that faces the arrangement in Fig. 1 is aligning the components, to insure a uniform light distribution along the matrix plane.

Waveguides can be used to build optical algebra processors. By using waveguides, the optical system can be made compact, and its alignment will be much easier than that of the astigmatic system. The distribution of the light across the waveguide is plotted in Fig. 2, which shows that the light is almost uniform along the waveguide. From the practical standpoint waveguides are more reliable to use in these systems than the astigmatic optics.

3.THE BOC DESIGN (HARDWARE AND SOFTWARE)

3.1 BOC HARDWARE DESIGN

The BOC hardware has three main parts as shown in Fig. 3. The optical system, the electronic circuit, and the digital processor. The optical system consists of the fully parallel matrix–vector multiplier. Light from the LED's representing .he $\underline{x}$ components are spread vertically by planner waveguides onto the columns of the matrix mask. The transmitted light is summed row wise by using another set of planner waveguides and detected by photodiodes which represent the output vector $\underline{b}$.

The electronic circuit acts as a feedback loop to correct for the input light of the LED's, until a solution is reached. The solution $\underline{x}$ will then be read and stored by the digital processor. Fig. 4 shows the electronic circuit used for the feedback loop.

The A/D and D/A conversion from and to the electronic circuit are performed by the digital processor.

3.2 BOC SOFTWARE DESIGN

The BOC software controls the Input/Output operations. Both the matrix A and the output vector $\underline{b}$ are read and stored by the digital processor. Tue vector $\underline{b}$ is then converted to analog voltage by a D/A converter, and it is assigned to the different ports of the electronic circuit. The analog optical processor solves for an approximate solution due to its inaccuracy. The digital processor reads and stores the approximate solution, $\underline{x}^{o}$ through the A/D converter, then it calculates the residue vector, $\underline{r}$, as,

$$\underline{r}=\underline{b}-A\underline{x}^{o}=A(\underline{x}-\underline{x}^{o})=A\Delta\underline{x} \tag{1}$$

Multiply Eq. (1) by a scalar s to make use of the whole dynamic range of the system, so Eq.(1) becomes,

$$s\underline{r}=A(s\Delta\underline{x}) \tag{2}$$

If the residue is not small enough, the system of linear Eq.(2) will be solved for $\Delta\underline{x}$ using the analog optical processor and,

$$\underline{x}^{1}=\underline{x}^{o}+\Delta\underline{x} \tag{3}$$

A new residue will be found for $\underline{x}^{1}$. The iteration process is continued by solving Eqs.(1) through (3) until a satisfactory solution is reached.

4.EXPERIMENTAL RESULTS

In this section we present the experimental results for solving a system of linear equations $A\underline{x}=\underline{b}$ using the BOC,where A,$\underline{b}$, and $\underline{x}$ are all positive.

The Log of the error and that of the residue are plotted versus the number of iterations. The error and the residue are defined as,

$$\text{Error}= ||(\underline{x}-\underline{x}^{k})||/||\underline{x}|| \tag{4}$$

$$\text{Residue}=||\underline{r}^{k}|| \tag{5}$$

Where $||.||$, is the Enclidean norm, $\underline{x}$ is the exact solution, $\underline{x}^k$ is the kth iteration solution, and $\underline{r}^k$ is the kth iteration residue.

Since we are dealing only with positive numbers in this paper, we used the absolute value of $\underline{r}$ to solve Eq.(2), then we set:

$$\underline{x}^{(n+1)}=\underline{x}^{(n)}+\Delta\underline{x} \tag{6}$$

when all the components of $\underline{r}$ are positive. And

$$\underline{x}^{(n+1)}=\underline{x}^{(n)}-\Delta\underline{x} \tag{7}$$

if all the components of $\underline{r}$ are negative. We reject the iteration when the components of $\underline{r}$ have different signs and take the previous one. By rejecting some iterations we are actually rejecting some corrections. This procedure slows down the convergence process.

In all the experiments performed, the iteration process is stopped when a 16 bit accuracy is reached. Fig. 5 shows that the BOC started with almost 30% error and it needed 6 iterations to converge to 16 bit accuracy. In Fig. 6(a) BOC started with almost 110% error, and the number of iterations needed was 21. Fig. 6(b) shows the Log of the residue as a function of the number of iterations. The fluctuations depicted by Figs. 6(a) and (b) is due to the rejection method used in the experiments.

4.1 EFFECT OF CALIBRATION

The analog optical system error is a major factor in the rate of convergence of the BOC. If that error is reduced, then the convergence is much faster. In order to illustrate this, the same problem has been solved twice with two different accuracies of the optical system. The analog optical system's error in the first time was 50%, and it was 30% in the second time. Twenty one iterations were needed by BOC to converge to the 16 bit accuracy for the first case. For the second case the number of iterations was reduced to 12. These results are plotted in Fig. 7.

4.2 RELIABILITY OF THE SYSTEM

System reliability for convergence have been tested and verified by solving the same problem several times, under different conditions. Results show that when the BOC is used, to solve a problem several times, the convergence rate will not be exactly the same for all the cases. However, the number of iterations needed by the BOC to converge to a certain accuracy is almost the same. Fig. 8 shows three different paths of convergence for the same problem. The BOC needed 13 iterations in the first run, 14 iterations in the second, and 11 in the third.

4.3 SOLUTION CONVERGENCE FOR THE SINGULAR MATRIX SYSTEM

Solving a system of linear equations with a singular matrix A is one of the problems that cannot be solved using conventional digital computer techniques. Singular matrices have a condition number equal to infinity, so their inverse does not exist, also they have infinite number of solutions. However, the BOC can be used to solve such systems [6]. The BOC converges much faster when A is singular, because a nonsingular matrix will have a unique solution. Due to the infinite solutions that a singular matrix has, the BOC produces different solution each time we try to solve the same problem again. Fig. 9 shows the BOC convergence for a singular matrix.

5.CONCLUSIONS

The BOC system was built and experimentally tested. The experimental results show great reliability of the processor in solving systems of linear equations. Overall 16 bit accuracy of the hybrid system was achieved with an analog optical system of 30% to 50% error. Higher accuracies of the solution can be obtained by increasing the number of iterations. The BOC also demonstrated to solve systems of linear equations with singular matrices.

We are considering in future work, bipolar numbers, complex numbers, and using SLM for the matrix mask.

6.ACKNOWLEDGEMENTS

This research was supported by the Innovative Science and Technology Office of the Strategic Defense Initiative Organization, administered through the Office of Naval Research under contract N00014–86–k–0591, and by Rome Air Development Center, administered through Georgia Institute of Technology under contract E–21–669–S23.

REFERENCES

1. H.J.Caulfield, J.H.Gruninger, J.E.Ludman, K.Steiglitz, H.Rabitz, J.Gelfand, and E.Tsoni, "Bimodal optical computers", Appl. Opt. 25(18),3123–3131, (1987).

2. W.K.Cheng and H.J.Caulfield, "Fully parallel relaxation algebraic operations for optical computers," Opt. Commun. 43,251 (1982).

3. M.A.G.Abushagur, and H.J.Caulfield, "Speed and convergence of the Bimodal optical computers", Opt. Eng. 26(1), 022–027 (1987).

4. M.A.G.Abushagur, H.J.Caulfield, and M.A.Habli, "Computer simulation of the Bimodal Optical Computer", Southeastern Computer Simulation Conference, SESC '87.

5. J.W.Goodman. A.R.Dias, and L.M.Woody, "Fully parallel, high–speed incoherent optical method for performing discrete Fourier transforms, "Opt. Lett. 2, 1 (1978).

6. M.A.G.Abushagur, H.J.Caulfield, P.M.Gibson, and M.A.Habli, "Superconvergence of hybrid optoelectronic processor", App. Opt. , Vol. 26, (23), 4906, 1987.

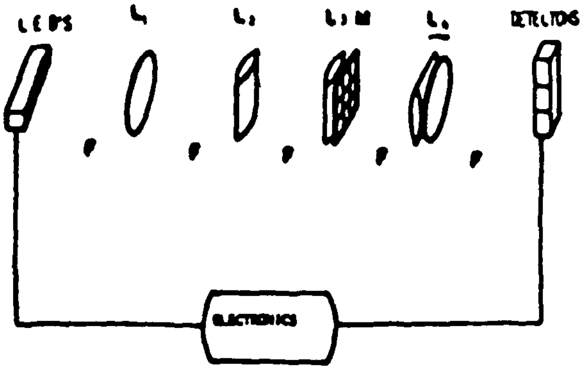


Fig 1. An astigmatic processor for solving system of linear equations.

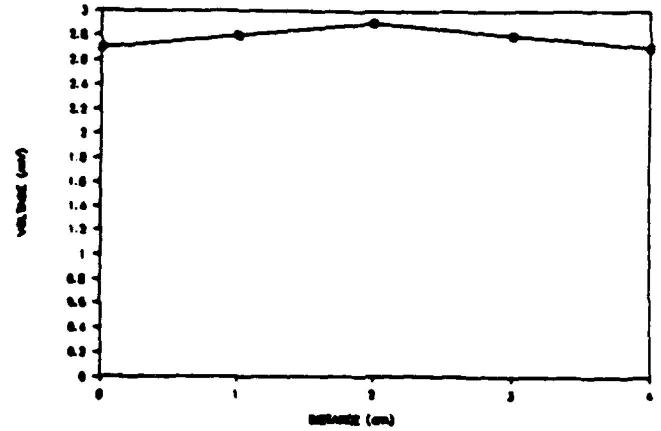


Fig 2. Light distribution across the waveguide.

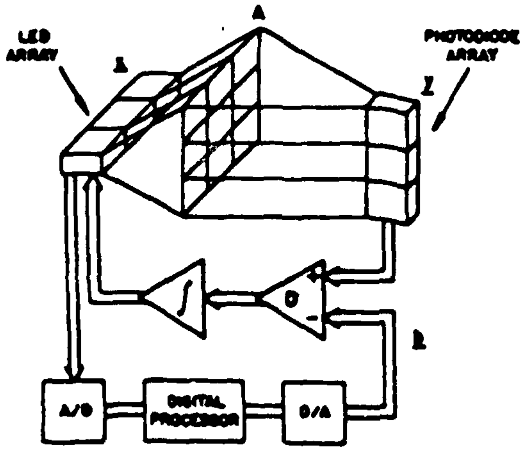


Fig 3. The Bimodal Optical Computer.

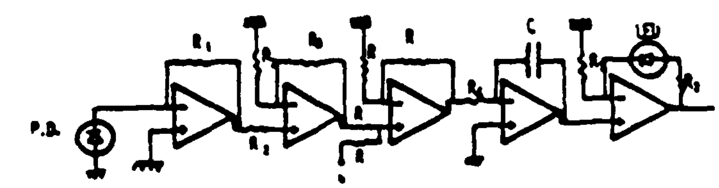


Fig 4. circuit diagram of the feedback loop for the optical system.

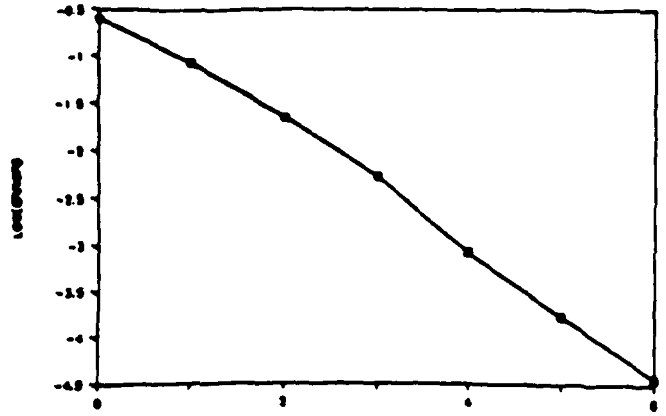


Fig 5. The Log(error) as a function of the number of iterations. The BOC started with 30% error.

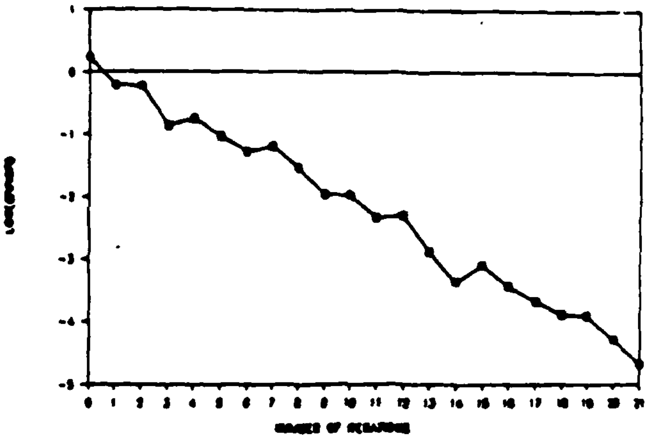


Fig 6(a). The Log(error) as a function of the number of iterations. The BOC started with 100% error.

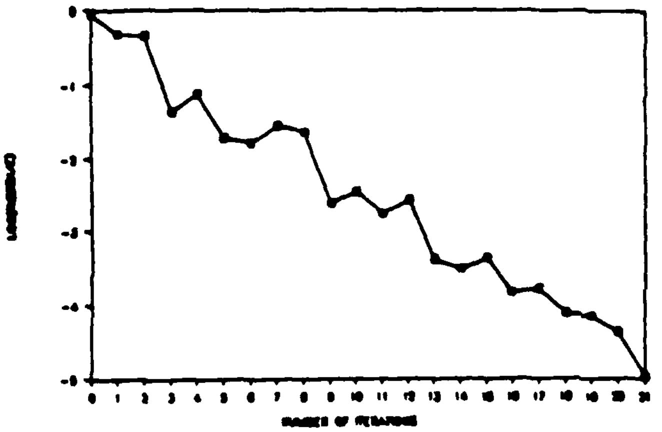


Fig 6(b). The Log(residue) as a function of the number of iterations.

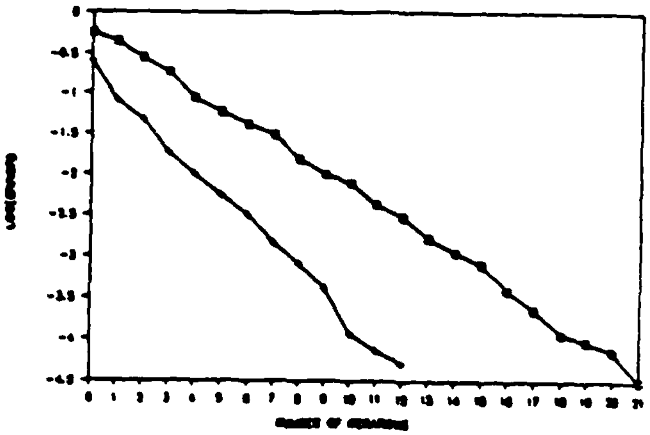


Fig 7. The Log(error) as a function of the number of iterations for the same problem, but with two different accuracies of the optical system.

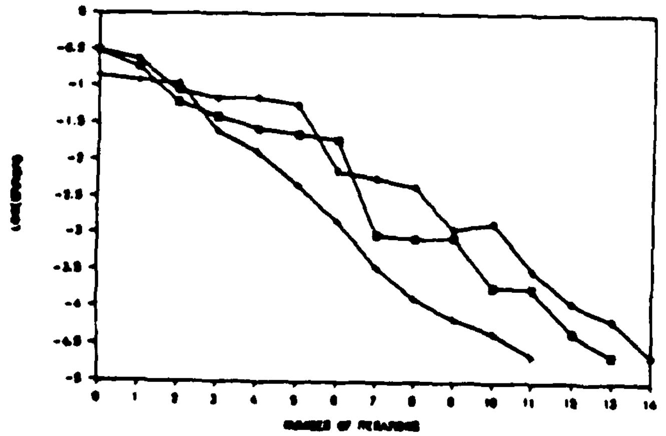


Fig 8. The Log(error) as a function of the number of iterations for the same problem done three different times.

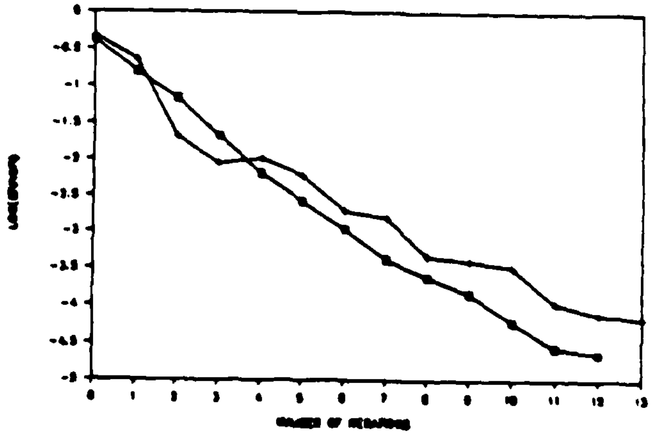


Fig 9. The Log(error) as a function of the number of iterations .The same problem done twice for a singular matrix A.

APPENDIX B

MASSIVE HOLOGRAPHIC INTERCONNECTION

The basic concept was described in late 1987 (Appl. Opt. 26, 4039). An extension soon followed (Lasers & Optronics, 1989). This, in turn, was followed by a detailed analysis of our concept and 1988 reinventions of it in the U.S., England, and Korea (Appl. Opt. 28, 311). A book chapter on this subject is now under preparation.

Reprinted from Applied Optics, Vol. *26*, page 4039, October 1, 1987

Parallel N^4 weighted optical interconnections

H. John Caulfield

University of Alabama in Huntsville, Center for Applied Optics, Huntsville, Alabama 35899.
Received 16 April 1987.
0003-6935/87/194039-02$02.00/0.
© 1987 Optical Society of America

While full optical interconnects of an $N \times N$ input signal array to an $N \times N$ output signal array through N^4 weighted interconnects is an important goal for optical artificial neural systems (ANSs), methods for doing this are rare. Goodman *et al.*[1] fully connected an $N \times 1$ array to a $1 \times N$ array. Sawchuk[2] has suggested a fixed N^4 interconnection method using replicated holograms for optical cellular logic. This works in principle but has extreme space–bandwidth requirements for large N. Sawchuk has described a 3-D dynamic interconnection network for interconnecting 2-*D* $N \times N$ arrays in parallel computing, but this network does not have arbitrarily variable weights.[3] I hope to show a simple optical N^4 interconnection method which uses only one noncritical lens, an $N \times N$ reflective spatial light modulator and a beam splitter as components.

It is convenient to think of the $N \times N$ input array as a matrix A with components a_{kl}. Likewise the output is an $N \times N$ array B with components b_{ij}. These are interconnected by a 4-D tensor T, i.e.,

$$B = \mathrm{TA}. \tag{1}$$

Equivalently,

$$b_{ij} = \sum_k \sum_l \mathrm{T}_{jkl} a_{kl}. \tag{2}$$

Let us donote oy T_{ij} the $N \times N$ array of T_{ijkl} elements arranged in the same way as the a_{kl} elements. That is, the tensor T can be thought of as N^2 different $N \times N$ weight arrays of the form T_{ij}, where T_{ij} is an $N \times N$ array of T elements needed for Eq. (2). Dropping the subscript kl from T_{ijkl} to T_{ij} is done for clarity in the following.

Figure 1 shows the basic scheme. The A matrix is inserted at the right side into the optical system via a reflective spatial light modulator (SLM). An $N \times N$ hologram array (which may be so large that it needs to be demagnified by relay optics before use as shown in Fig. 1) is illuminated by a reconstruction beam and provides the $N^2\mathrm{T}_{ij}$ arrays. In Fig. 1, we see that the T_{ij} arriving at the reflective SLM are N^2 products of the form $\mathrm{T}_{ijkl}a_{kl}$. These are collected in the B plane (the image of the hologram array).

In practice it may be necessary to make minor modifications on the apparatus of Fig. 1. The hologram array, SLM, and detector array may be of different sizes. This requires relay lenses to magnify or demagnify one or more of these to *achieve the* Fig. 1 *configuration* or some simple variant of it. For example, the hologram array may be quite large. Assuming a 2-mm diam hologram to store a 1000×1000 T_{ij} mask, we need a 2- × 2-m hologram array to store 10^{12} weights. This certainly precludes some uses. A $500 \times 500\mathrm{T}_{ij}$ mask needs a 1-mm hologram, and we only need a 50- × 50-cm array to store $(500)^4 = 67.25 \times 10^{10}$ weights. Fresnel diffraction considerations make it desirable to keep the holograms larger than or equal to . mm. Thus if we drop to a $128 \times 128\mathrm{T}_{ij}$, we need a 12.8- × 12.8-cm array to store the $(128)^4 = (2^7)^4 = 2^{28} \times 2.5 \times 10^8$ weights.

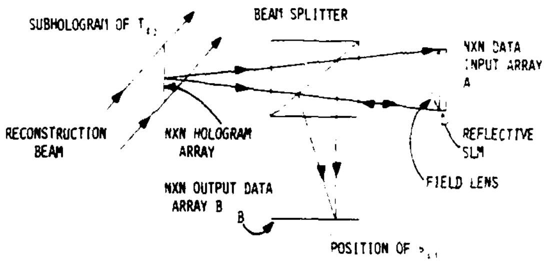


Figure 1. Configuration of a parallel optical interconnection between an $N \times N$ *input array and* A *and an* $N \times N$ *output* B. *An* $N \times N$ *array of holograpms each containing and* $N \times N$ *mask array is illuminated in parallel to produce the* N^4 *interconnection.*

To record each subhologram we must reverse Fig. 1. A point source at the i, j position in the B plane illuminates the SLM. The T_{ij} pattern is written onto the SLM. A coherent reference beam conjugate to the Fig. 1 reconstructing beam allows the subhologram to be recorded.

Optical parallel N^4 interconnections are seen to be quite straightforward. No technology breakthroughs are required to achieve $N = 10^3$ or $N^4 = 10^{12}$. Recording the master hologram as a whole or in parts may prove slow, but mass-produced copies can be made quickly and inexpensively.

This work was sponsored primarily by the Department of the Navy under contract N00014-86-K-0591.

References

1. J. W. Goodman, A. R. Dias, and L. M. Woody, "Fully Parallel, High-Speed Incoherent Optical Method for Performing Discrete Fourier Transforms," Opt. Lett. 2, 1 (1978).
2. B. K. Jenkins and A. A. Sawchuk, "Optical Cellular Logic Processors," J. Opt. Soc. Am. A 2(13), P26 (1985).
3. A. A. Sawchuk, "3-D Optical Interconnection Networks," in *Proceedings, Fourteenth Congress of the International Commission for Optics*, 22–24 Aug. 1987, Quebec, Canada (to appear).

NEWS

RESEARCH

A Breakthrough For Optical Neural Nets

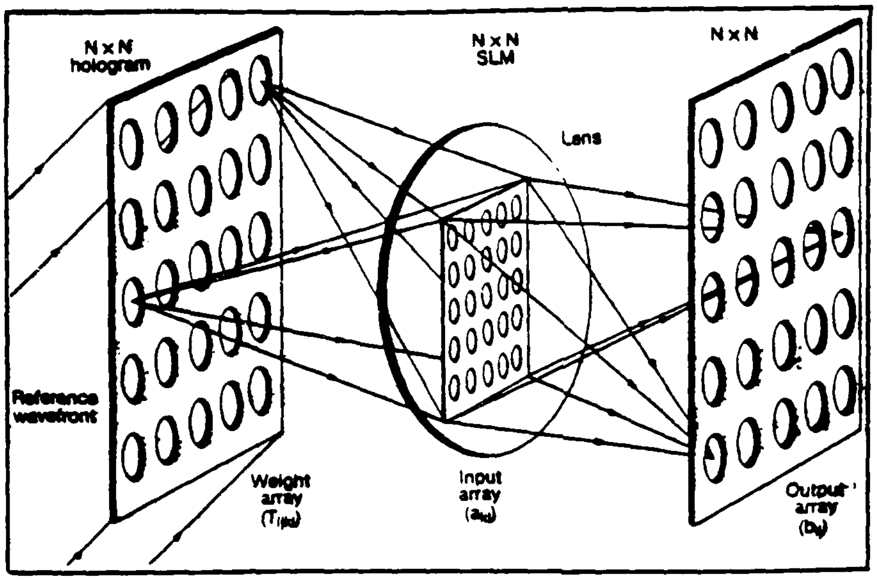


Figure 1. One trillion weighted, parallel interconnections for optical neural-net computing tasks can be accomplished with the above design. Data input is achieved with an N × N SLM input array. Each element of the input array is combined with an N × N weighted array from an N × N holographic array into an N × N output array.

Since the advent of the laser, a quiet struggle has been going on between optics and electronics. For a century it seemed that Maxwell's equations were the only laws these two fields had in common. Now, however, diode lasers and thousands of kilometers of fiberoptic cable are staging a virtual takeover in telecommunications, erasable optical disks are edging in on conventional magnetic storage, and even the hallowed halls of electronic computers are being infiltrated by new optoelectronic techniques.

The latest optoelectronic coup is being staged by H.J. Caulfield, director of the Center for Applied Optics at the University of Alabama in Huntsville. Caulfield has put forth an "existence proof" that he says shows, "that there is a vital task in computing that optics can do *now* and electronics can *never* do." What's more, he says it can be done with technology that has been around for twenty years. If true, this is the breakthrough that optical computing has been waiting for. As Caulfield puts it, "I think what we're seeing is the real birth of optical computing."

In an interview with *L&O* on August 19, Caulfield revealed that optical techniques offer the only possible solution to the massive, parallel, weighted interconnections of neural networks. Neural networks, or "neural nets," form the foundation of a computer architecture designed to mimic the human brain by forming millions, even trillions, of individual, parallel interconnections. As with neurons, all these interconnections could be individually weighted and connected to an equal number of outputs. Such an architecture was developed by Warren McCulloch as long ago as the mid 1920s, but has only recently been studied as a possible solution to highly complex, repetitive computing problems requiring high-speed solutions, such as pattern recognition.

> "I think what we're seeing is the real birth of optical computing."

Caulfield used *reductio ad absurdum* to prove the futility of making 10^{12} parallel, weighted interconnections electronically. He explained that since electrons interact with one another, the connections would all have to be made with individual "wires" or electron carriers. Submicron carriers on silicon chips have just become possible, so it is not inconceivable that such carriers could be made and packed together as closely as 1 micrometer by the next century. For 10^{12} connections, the wires would have to be packed together in a two-dimensional array 1 × 1 meter, and then rearranged somehow to form the interconnections, which, according to Caulfield, would require wire lengths of 10 meters or so. That leads to a 10-m^3 conductive mass with bothersome inductance and crosstalk.

Furthermore, even if the conductivity (weights) of all these wires could be set independently with no space, time, or cost penalty, each wire must be connected to an input and an output. When added to the interconnections, this leaves 4×10^{12} attachments to make. If 4,000 connections could be made every second, it would take 10^9 seconds to complete the task, which adds up to something over 25 millennia.

In a paper submitted to *Applied Optics*, Caulfield suggests an optical method of accomplishing the same thing. The technique makes use of holographic technology from the 1960s—page-oriented holographic memories. Figure 1 shows how it works.

To produce 10^{12} optical interconnections, input data is encoded onto a $10^3 \times 10^3$ array in the form of a transmissive spatial light modulator (SLM); although a reflective SLM can also be used. Each element of the the SLM input array can be assigned a set of weighted values by means of a large holographic array.

The holographic array is 10^3 holograms high and 10^3 holograms wide. Each of the elemental holograms in this array is made in such a way as to produce a $10^3 \times 10^3$ pattern onto the $10^3 \times 10^3$ SLM input array. So the holographic array responsible for assigning the weighted interconnections can be thought of as N^2 different N × N weight arrays, and can be represented by a four-dimensional tensor T_{ijkl}.

When the weighted arrays of holograms are reconstructed with a reference beam and imaged onto the $10^3 \times 10^3$ SLM input array, a $10^3 \times 10^3$ output array, B, is produced. B = TA and has elements

$$b_{ij} = \sum_{k,l} T_{ijkl} a_{kl}.$$

A 2 × 2-meter holographic array consisting of $(10^3)^2$ holograms, with each 2-mm-diameter hologram storing a $10^3 \times 10^3$ weight array, would yield 10^{12} weighted interconnections when combined with the $10^3 \times 10^3$ SLM input array.

It is impossible to fathom the effects this development will have on optical neural-net design, but Caulfield has managed to combine mindboggling complexity with stupefying simplicity. He suspects that the most time-consuming part of his optical neural-net design will be learning how to weight the input array for each optical computing task. This, he believes, could take months to years and require the learning capacities of traditional electronic computers. Once a master weight array is produced, however, neural-net operation time should be in the realm of milliseconds, and successfully "programmed" master holograms could be cloned in seconds for mass production.

"The role of electronics, with its great flexibility and accuracy, is learning. The role of optics is doing what electronics learns," says Caulfield. "Brains use the same equipment for both tasks, but why should we? We should let optics and electronics do what each does uniquely well. The war between optics and electronics is a foolish one. They each have a major role. What we have done is [to] show, in one vital area, what those roles are."

—Tom Higgins

Can Superconductors Replace Fiberoptics?

Could the new high-temperature superconductors provide much greater transmission bandwidth than fiberoptics for long-distance communications? The recent demonstration that such materials can transmit picosecond pulses has some observers believing so. However, others are more cautious, noting that the two key experiments sent picosecond pulses through only five-millimeter lengths of thin-film superconductor.

> The experiments are clear indication of high potential bandwidth for the new superconductors.

The two experiments were announced nearly simultaneously. One was by a team from the University of Rochester and Cornell University, the other by a team at the IBM T. J. Watson Research Center in Yorktown Heights, N.Y. Both were working with thin films of yttrium barium copper oxide, the best-known member of the family of new materials that are superconducting at temperatures of 90 or 100 Kelvin. Both passed picosecond-domain electrical pulses through 5-mm lengths of superconducting film and could detect no pulse distortion or dispersion over that scale. Both involved researchers well known for their work on ultrashort optical pulses.

However, the two experiments differ in detail, and the two groups differ radically in how far they are willing to extrapolate the results. Gerard Mourou, director of the ultrafast science center at Rochester's Laboratory for Laser Energetics, predicted, "over distances of miles, lossless superconducting transmission lines with 100 times the capacity of optical fiber systems could be developed." Much more cautious was Alex Malozemoff, research division coordinator for superconductivity at IBM, who said, "I don't think we've yet tested the superconductors in a regime to talk about long-distance transmission."

The thin-film superconductor used in the Rochester-Cornell experiments was deposited on a zirconium-oxide substrate by Robert Buhrman, professor of applied and engineering physics at Cornell. The submicrometer film was etched to form a

Reprinted from Applied Optics, Vol. *28*, page 311, January 15, 1989

Massive holographic interconnection networks and their limitations

Joseph Shamir, H. John Caulfield, and R. Barry Johnson

Fundamental and practical limitations to be encountered in the implementation of massive free space optical interconnects are discussed in detail, and some improved architectures are proposed. The long term optimum design uses currently unavailable large arrays of laser diodes. An interim solution, using available spatial light modulators, is shown to be capable of storing $\sim 10^{10}$ bits of information and performing $\sim 10^{11}$ interconnections/s.

I. Introduction

There is an increasing interest in massive optical interconnection networks for incorporation in communication and signal processing systems.[1-6] Optical architectures are particularly attractive for the implementation of interconnection networks with extremely high complexity that are impractical with conventional electronic systems. Neural networks[7-22] that are based on massive weighted interconnections are good examples of such systems. Many of the architectures considered in the above-mentioned references employ the extensive interconnectivity available in free space propagation of light waves. Only a few of these publications have, however, seriously discussed the actual feasibility of large scale implementation.[2,10,11] A more common attitude is the description of a system architecture with a statement on the expected performance. Sometimes a demonstration is presented with a small array of input data, but, in many cases, the limitations imposed on the upscaling possibilities are ignored. Several limitations stem from fundamental physical processes such as diffraction and coherence, while others are due to technical difficulties such as the angular dependence of spatial light modulators (SLMs) and the actual shift variance of real spatial filters.[23]

The main objectives of this work are the analysis of the degradation factors that limit the performance of practical interconnection networks and the derivation of fundamental and technical constraints on the realization of an actual physical system. The analysis is based initially on a recently proposed architecture,[7] and two additional configurations are introduced with an attempt to overcome some of the difficulties. Although we treat here a very specific application, the results are relevant to a considerable variety of other optical processing architectures that have been proposed in the past or will be proposed in the future.

In the next two sections, we describe the anticipated performance of an ideal system disregarding all constraints that will be analyzed in Secs. IV–VII in detail. Part of this analysis is based on a more exact mathematical description of the whole process given in Appendix A for a scalar paraxial approximation. For most parts of the analysis we assume that the system must perform all possible interconnections among all the channels available and base the calculations on worst case conditions. It is to be expected that if these worst case conditions are replaced by some statistical average, the derived constraints may be appreciably relaxed. Furthermore, in many applications one does not need all the possible interconnections, and then the system may be divided into subapertures. In that case our estimated constraints relate to the largest subaperture and the complete system may become much larger.

Two new architectures are introduced in Sec. VIII with an attempt to reduce the constraints derived for the original architecture. In Sec. IX we perform a general analysis for the derivation of the laser power requirements, and in Sec. X a detailed case study is given with the derivation of design parameters for an actual system that may be implemented in practice with presently available devices. This system consists of 256 × 256 input and output arrays with 256^4 weighted interconnections capable of performing 10^{11} interconnections/s. Important concluding remarks, related to the actual implementation of an interconnection network, are given in Sec. XI.

The authors are with University of Alabama in Huntsville, Center for Applied Optics, Huntsville, Alabama 35899.

Received 2 March 1988.
0003-6935/89/020311-14$02.00/0.

II. Ideal Performance of Basic Architecture

The basic configuration, shown in Fig. 1, is the transmissive version of the reflective system described in Ref. 7. It consists of a hologram array of linear dimensions H containing $N_h \times N_h$ holographic optical elements, an SLM of size S with $N_s \times N_s$ pixels sandwiched between two lenses with respective focal lengths f_1 and f_2, and a detector array D with $N_d \times N_d$ detector elements. The ijth hologram in the array is imaged by the double-lens configuration onto the ijth element of the detector array. This hologram diffracts light from a reconstruction beam with an efficiency t_{ijkl} toward the klth pixel in the SLM. The same pixel receives a weighted fraction of the light diffracted also from all other holograms, but, assuming a linear interaction in the SLM, these are separated again on arrival at the detector array. Thus, ideally, each detector receives the sum of all the weighted beams just from a single hologram element. Mathematically, if the power transmittance of the klth pixel in the SLM is a_{kl}, the total power received by the ijth detector will be

$$b_{ij} = \sum_{kl} t_{ijkl} a_{kl}, \qquad (1)$$

where, for the time being, coherence effects have been ignored.

Performance degradation due to coherence is just one of the factors that is discussed later along with some other effects that limit the scale-up capability of this and many other architectures. This system in its ideal form may be viewed either as a matrix–matrix multiplier of a 4-D matrix by a 2-D matrix or as a vector–matrix multiplier with vectors of $N_s \times N_s$ dimensions,

$$\|T\|\mathbf{A} = \mathbf{B}. \qquad (2)$$

The elements of the input vector (or matrix) are introduced by the transmittance of the SLM pixels with the hologram providing the fixed matrix $\|T\|$. The output vector is read out from the detector array.

Alternatively, we may consider this an interconnection network with N_s^2 channels that are interconnected by $N_s^2 \times N_h^2$ weighted interconnections that are hardwired for a given hologram array.

III. Hologram Recording

To implement the above architecture one must also devise a system for recording the required large hologram array. Within the present state of art the practicality of computer generation with electron-beam writing appears to be out of the question for these large arrays. Thus one must resort to optical recording, preferably with computer assistance.[10,11] Several procedures may be envisioned for the implementation of the hologram recording process. The most obvious of these processes is based on the same optical system as the interconnection network itself (Fig. 1) where each element of the detector array is replaced one at a time by a point source. A useful realization of this point source may be the endface of a single-mode optical fiber that can be easily positioned and aligned with a computerized robotic arm. To record the ijth hologram this source is positioned at the location of the ijth detector pixel, oriented for optimal illumination of the SLM, and imaged onto the hologram by the two lenses. The SLM, sandwiched between the lenses, writes the desired interconnection pattern. This special lens configuration is useful to keep all incident ray angles on the SLM constant for a given hologram allowing for adjustments to take care of the angular variation of the SLM transmission characteristics. The constraints related to the operation of SLMs are discussed in more detail later. To attain small repeatable high quality holograms, a random phase plate over the SLM may be useful[24] as discussed further in Sec. VI. The overall process of recording and reconstruction is mathematically evaluated in Appendix A within the paraxial approximation for an ideal case using operator notation.[25–27]

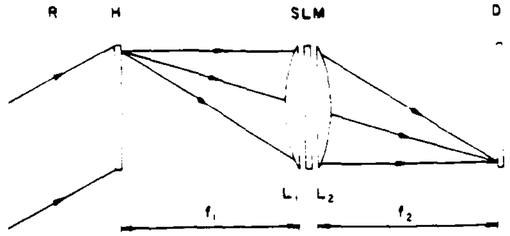


Fig. 1. Basic configuration for an N^4 interconnection network: H, hologram array illuminated by a reconstruction beam R; SLM, spatial light modulator between two lenses L_1 and L_2 with their respective focal lengths f_1 and f_2; D, detector array or an array of nonlinear optical devices.

In the above recording configuration it was assumed that an oblique reference beam, conjugate to the one indicated on the figure, is incident on the hologram. Alternatively, one may use a point source reference situated on the optical axis at the SLM plane. This will allow an axial reconstruction beam resulting in a reduced bandwidth requirement for the holograms and a simpler reconstruction configuration. The penalty for these benefits is removal of the central portion of the SLM and the introduction of aberrations induced by spherical-wave recording and reconstruction.

IV. Semiquantitative Constraint Estimation

The exact analysis of the physical processes involved in the operation of the proposed architecture is quite complicated and outside the scope of this work. Nevertheless an appreciable insight can be obtained by evaluating the diffraction effects in the scalar paraxial approximation. In Appendix A we present a Fourier optics description of the complete process starting from the hologram recording stage and ending at the detection of the output vector. Keeping in mind the results in the Appendices, in the present section we use a somewhat different approach that allows us to take into account in a semiquantitative way effects induced by off-axis propagation.

To obtain an estimate on the limitations imposed on the system of Fig. 1 we consider first the diffraction effects occurring while light is propagated from each hologram in the array toward the various pixels of the SLM. For proper performance we must require that most of the light addressing a pixel in the SLM should be incident on this pixel, and only a negligible fraction will be spilled over to other pixels.

We denote the linear dimensions of the respective pixel sizes in the hologram array, the SLM and the detector array by p_h, p_s, and p_d, and the respective center-to-center distances of the pixels by d_h, d_s, and d_d. As discussed later, the center-to-center distances are not necessarily equal to pixel sizes. With the parameters defined earlier we have the relations

$$N_s = S/d_s; \qquad N_h = H/d_h; \qquad N_d = D/d_d. \tag{3}$$

If we assume diffraction-limited performance with circular hologram elements, we may require that d_s be not smaller than the diameter of the first Airy disk obtained by focusing an aperture, the size of the hologram, onto the SLM plane. That diameter is

$$a = 2.44\,\frac{\lambda r}{p_h}, \tag{4}$$

where $r = f/\cos\theta$ is the distance between the ijth hologram and the klth SLM pixel (see Fig. 2). We also must consider the elongation of the spot due to the inclination of the beams incident on the surface of the SLM. In principle we should also consider the $\cos^4\theta$ flux density falloff, but we assume that this may be precompensated for during the hologram recording stage by modifying the assigned weights.

The optical configuration with the hologram in the focal plane of the lens ensures that all beams from a single hologram are incident on the SLM approximately at the angle at which the central pixel is addressed. Thus from Fig. 2 it is evident that the maximal value of this angle $\theta_{\max}$ is obtained for the hologram situated at the corner of the array and is given by

$$\tan\theta_{\max} = \frac{H}{\sqrt{2}f_1}. \tag{5}$$

All beam spots for this marginal hologram will be elongated by a factor $1/\cos\theta_{\max}$. In calculating the distance r, larger angles should be also considered, but, to reduce the algebraic complexity, we take into account only an average distance traveled by the various beams emerging from this hologram keeping in mind that the actual situation is worse. For this average distance we may put $r \approx f_1/\cos\theta_{\max}$. With all these considerations a minimal requirement for pixel separation is given by

$$d_s \geq 2.44\,\frac{\lambda f_1}{p_h \cos^2\theta_{\max}}. \tag{6}$$

Solving Eq. (5) for f_1 and substituting into Eq. (6), we obtain

$$d_s \geq 2.44\,\frac{\lambda H\sqrt{2}}{p_h \sin 2\theta_{\max}} \tag{7}$$

$$\frac{d_s}{\lambda} \geq \frac{3.45}{\sin 2\theta_{\max}}\,\frac{H}{p_h}. \tag{8}$$

This relation may be regarded as the constraint set by the requirement of diffraction-limited performance.

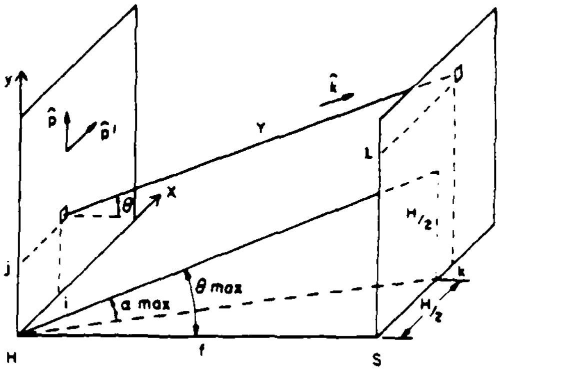


Fig. 2. *Definition of geometrical parameters:* H, hologram plane: S, SLM plane or the detector plane in the modified architecture: r, distance between the ijth hologram and klth SLM pixel. Polarization vectors $\hat{P}$ and $\hat{P}'$ as well as the angle $\alpha_{\max}$ and propagation vector $\hat{k}$ are discussed later.

In its present form one may interpret it as a limitation of the ratio H/p_h for a given SLM (that is, a given value of d_s) operating at a given wavelength. From the optical designer's point of view, $\theta_{\max}$ is determined by the numerical aperture of the optical system. However, as shown in later sections, additional constraints should be considered. As a demonstrative example to the meaning of relation (8) we assume the unlikely angular limitation $\theta_{\max} = 45°$ (corresponding to an f/No. of 0.7) and take $\lambda = 0.5$ μm, we need a pixel center-to-center distance, $d_s = 173$ μm, to obtain a ratio $H/p_h \leq 100$. This means that in these conditions one is limited to a hologram array of $N_h \times N_h = 100 \times 100$ elements unless holograms are allowed to overlap spatially, or alternatively an appreciable amount of crosstalk is allowed. The number of elements in the SLM, however, is not limited by relation (8), and if we want to implement a system with input vectors of rank $N_s \times N_s = 1000 \times 1000$ we need an SLM with $S = 17.3$ cm.

The limitations on the absolute size of the holograms may be determined by considering the requirements for space–bandwidth product or rather a quantity that we shall call information content (IC). If the resolution of the holographic material is $1/l$ and it can record g_h distinct gray levels, we obtain the approximate value

$$\mathrm{IC}_h = \left(\frac{p_h}{l}\right)^2 g_h. \tag{9}$$

If the SLM has a gray level capability of g_s levels, its IC is

$$\mathrm{IC}_s = N_s^2 g_s, \tag{10}$$

which should satisfy the relation $\mathrm{IC}_h \geq \mathrm{IC}_s$ leading to

$$\frac{p_h}{l} \geq N_s \sqrt{\frac{g_s}{g_h}}. \tag{11}$$

It should be noted here that, apart from material limitations, l is also limited by the recording wavelength: a holographic grating can never have its interference pattern with spatial frequency higher than $2/\lambda$, and in most recording configurations one has $l > \lambda$. Continuing with our previous example with $N_s = 1000$, taking the limiting value $l = 0.5\ \mu$m, and assuming $g_h = g_s$, we obtain $p_h \geq 500\ \mu$m. Usually we shall need at least twice this value to incorporate the hologram carrier frequency for off-axis reconstruction. If a random phase modulator is attached to the SLM, as proposed in Ref. 24, the IC_s increases, which necessitates an even larger hologram.

This first-order estimation leads to severe limitations on the number of elements that may be processed in parallel using a single optical system. A more detailed analysis is required to derive trade-offs applicable to specific system design. For example, one may consider partially overlapping of holograms to obtain a larger number of holograms while still satisfying the restriction on the ratio between hologram size and array size [Eq. (8)]. One may also relax the crosstalk limitations that will influence relation (6). On the other hand, we must consider the deteriorating contribution of coherence effects, aberrations, scattered likgth, and performance limitations of SLMs.

V. Crosstalk Considerations

In the configuration of Fig. 1, crosstalk occurs on the SLM plane and also on the detector plane. The light from a single hologram is split into several beams of various intensity, each of which is ideally focused into a separate pixel of the SLM. These beams are modulated by the SLM and then converge to a single-detector element. The crosstalk on the SLM plane originates mainly from light injected into the klth pixel from matrix elements $t_{ijk'l'}$ addressed to different pixels with $k'l' \neq kl$. As long as the interaction in the SLM is linear, the mixing of light from several holograms has no effect at this plane but becomes important on the detector plane where light originating from one hologram leaks through to unintended detector elements.

To consider crosstalk over the SLM plane, we denote by ϵ_s the power originating from a certain hologram that may be incident on a single pixel due to leakage from other beams that are not supposed to contribute to this pixel. One may state that the weight attributed to this element for a certain interconnection is increased by this value. However, since ϵ_s usually has a statistical nature, we may consider it as the error assigned to the element of the interconnection matrix. Thus, instead of having a well-determined weight multiplying each SLM element, we must include some average bias level $\epsilon_s/2$ and write

$$t_{ijkl} \rightarrow t_{ijkl} + \frac{\epsilon_s}{2}(1 \pm 1). \qquad (12)$$

T̓ s error has several contributions that include diffraction, aberrations, inclination factors, scattered light, and coherence effects.

A similar effect may be observed on the detector plane where ideally the light emerging from each hologram should be focused into a single pixel. Denoting the contribution of light power from other holograms by ϵ_d we obtain the value of the vector elements with error,

$$b_{ij} = \sum_{kl}\left[t_{ijkl} + \frac{(\epsilon_s)_{kl}}{2}(1 \pm 1)\right]a_{kl}/2 + \frac{(\epsilon_d)_{ij}}{2}(1 \pm 1), \qquad (13)$$

where we noted the dependence of the error terms on location and took into account the bias terms due to the noise. Naturally these terms also depend on the matrix and vector elements themselves. Thus the above relation is essentially nonlinear in the matrix and vector components. Thus one may only estimate some maximal error values and possibly determine their statistical nature for a given situation.

According to our design objectives, the dimensionality of the detector plane should be equal to the dimensionality of the hologram plane (that is, $N_d = N_h$ although not necessarily equal to N_s). Since our optical system is essentially an imaging system between the detector and hologram planes, we have the geometrical relations

$$d_d = \frac{f_2}{f_1} d_h, \qquad (14)$$

$$D = \frac{f_2}{f_1} H. \qquad (15)$$

In principle, we also have the relation

$$p_h = \frac{f_2}{f_1} p_h, \qquad (16)$$

where p_h' is the image of p_h over the detector plane. However, because the reconstructed wavefront over p_h is the phase conjugate of the writing beam, the distribution within p_h will be quite nonuniform. As a matter of fact, if p_h is very large and the SLM has unit transmittance, the complete reconstructed wavefront will be concentrated into the region occupied by the source during hologram writing. As is evident from the paraxial calculation in Appendix A, if we have the SLM in place, according to Eq. (A25), the power distribution will approximately be (ignoring coherence effects to be discussed in the next section) that of a sinc^2 function, the extent of which is determined by the SLM pixel size. If the hologram has a finite size, this distribution will be widened by a convolution containing the window function as derived in Eq. (A27). If this window function is not too small, that is, it satisfies a relation of the form (8), we may state that the crosstalk over the detector plane is generally proportional to the diffraction spot size of the SLM pixel over this plane. Using considerations similar to those leading to Eq. (6), we may write for the power that spills over the area of the detector pixel

$$\epsilon_d \propto \left(\frac{\lambda D}{d_d p_s \sin 2\theta_{\max}}\right)^2, \qquad (17)$$

where $\theta_{\max}'$ is the maximal angle in the detector plane. If we define this angle in a similar way to the definition of $\theta_{\max}$, we have here too a geometrical relation

$$\tan\theta_{\max} = \frac{f_1}{f_2}\tan\theta_{\max}. \qquad (18)$$

The above discussion indicates that the crosstalk term at the detector plane and, according to the discussion in the previous section, also the crosstalk term on the SLM plane are both inversely proportional to the SLM pixel size. Thus, to minimize crosstalk due to diffraction one would like to increase p_s as much as practically feasible. However, if we intend to limit the SLM pixel size to its minimal value according to Eq. (6), we may substitute the equality relation of Eq. (8) into Eq. (17) to obtain

$$\epsilon_d \propto \left(\frac{p_h d_s f_2 \sin 2\theta_{max}}{d_d p_s f_1 \sin 2\theta_{max}}\right)^2 , \tag{19}$$

where we used Eq. (15). For the crosstalk over the SLM we may write a similar expression:

$$\epsilon_s \propto \left(\frac{\lambda H}{d_s p_h \sin 2\theta_{max}}\right)^2 \cdot \tag{20}$$

Multiplying the last two equations we may write

$$\epsilon_s \epsilon_d \propto \left(\frac{f_2}{d_d p_s f_1 \sin 2\theta_{max}} \lambda H\right)^2 \cdot \tag{21}$$

For relatively small angles we may also write with the help of Eq. (18)

$$\epsilon_s \epsilon_d \propto \left(\frac{f_2^2}{f_1^2 d_d p_s \sin 2\theta} \lambda H\right)^2 \cdot \tag{22}$$

This relation is quite interesting as it indicates some possibilities for improving system performance by increasing the SLM and detector pixel sizes and the ratio f_1/f_2. Unfortunately, these parameters cannot be adjusted independently, and they must be considered together in some optimization process.

To evaluate the order of magnitude of the crosstalk errors we follow the analytical results of Appendix A with some numerical calculations assuming rectangular pixels as shown in Fig. 3. To evaluate the crosstalk over the detector plane we may start from Eq. (A25) and keep in mind that a similar procedure applies also for the SLM plane. We may normalize the argument of the sinc function to the value of its argument at its first zero:

$$x_0 = \frac{\lambda f}{2p_s} , \tag{23}$$

where a detector pixel size

$$p_d = 2x_0 = \frac{\lambda f}{p_s} \tag{24}$$

covers the whole central lobe of the sinc function. It should be noted that this size is smaller than the one used in the previous estimations.

If we illuminate the detector array with such a sinc function centered on a detector denoted by A in Fig. 3, 'the detected intensity will be the integrated square of the sinc function over the area of each detector element. The curves in Fig. 4 are the calculated integrals over pixels situated relative to A as the ones denoted by B. The values are given as the percentage relative to the integral over pixel A as a function of the normalized pixel separation d_d/p_d for three normalized values of p_d with the middle curve ($p_d = 1$) corresponding to the value given by Eq. (24). As expected, the crosstalk increases drastically if the interpixel distance drops below the size of the central lobe, but there is an appreciable crosstalk even for quite large separation. The curves indicate that we do not gain much by enlarging the detector elements much above $2x_0$. Assuming that this is a good choice for the dimension of the active part of the detector ($p_d = 2x_0$), we obtain a calculated value for the crosstalk between two adjacent detectors of 3.5% if they touch each other ($p_d = d_d$). This value for d_d is technically not feasible, and we rather take $d_d = 4x_0$ with which we obtain a crosstalk of 0.74%. The crosstalk to a more distant pixel (D in Fig. 3) with its center at $8x_0$ is ~0.18%, while the value for the nearest diagonally positioned pixel (C in Fig. 3) is only 0.075%. Assuming this arrangement we observe a maximum of four pixels each contributing 0.74%, four pixels contributing 0.075%, and four more contributing ~0.18%, each. Taking into account the smaller contributions from more distant pixels, a maximum estimated crosstalk value is ~4%.

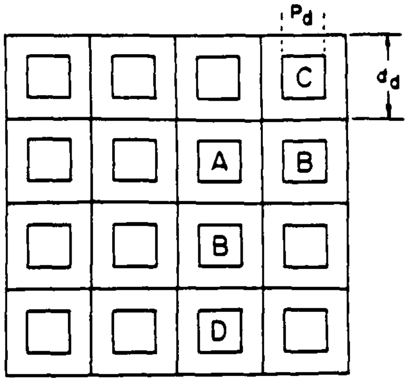


Fig. 3. Layout of detector array.

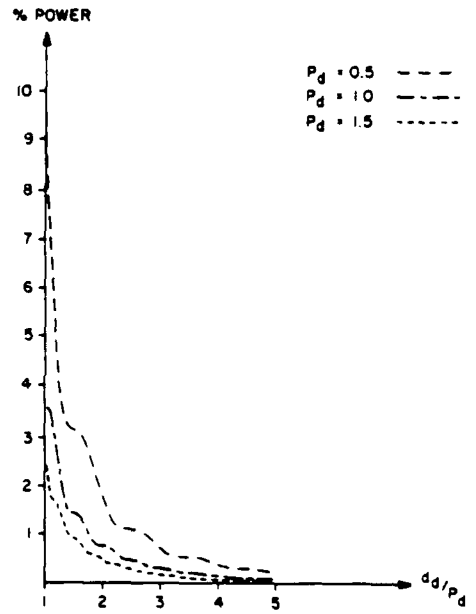


Fig. 4. Crosstalk percentage as a function of relative center-to-center distance of detectors. Parameter is detector size relative to central lobe o sinc function.

As we have seen, due to the special structure of the diffraction pattern from a rectangular aperture, the crosstalk between diagonally positioned pixels is about an order of magnitude smaller than between adjacent ones. Consequently, a checkerboard configuration may be advantageous for some applications even if this requires a fourfold reduction in the number of pixels. Naturally, even in this case we cannot forget the second neighbors D that still contribute 0.18%, each leading to ~1% combined contribution from the eight nearest pixels.

The above last number for the fractional error appears quite affordable, but we may run into difficulties even with this quantity. In general, we may define a quantity ϵ_f that gives the fractional leakage from one pixel addressed by a single channel into a neighboring one. If the maximum possible power arriving to the detector from a single channel is I, the crosstalk to a neighboring pixel will be $I\epsilon_f$. If we have M such pixels around each other pixel ($M = 8$ in the above example), we obtain a maximum crosstalk from these pixels of $MI\epsilon_f$. Assuming that this contribution is the most significant and we may neglect other contributions, we shall obtain the maximal value of the crosstalk when all N_s^2 channels are addressed with full weight to all the neighboring pixels:

$$(I\epsilon_d)_{\max} \approx MN_s^2 I\epsilon_f, \qquad (25)$$

which may become much larger than I even for small ϵ_f. We shall return to this subject and discuss it further in relation to the estimation of laser power requirements.

All the above considerations presume ideal performance and alignment. One essential technical factor to treat is proper alignment. If we have a misaligned pixel it will be shifted on the curves of Fig. 4. For example, according to the calculations with the above configuration, the measured power for a pixel displaced in one direction by 5% will be off by 1%.

VI. Coherence Effects

In Eq. (1) we assumed that the power contributed by the different pixels in the SLM is combined incoherently at a detector pixel. As is quite evident, this assumption is incorrect since the holograms are illuminated by coherent light and one must consider coherent superposition. To do this we have to evaluate the complete complex amplitude distribution at the detector plane. As a first-order approximation for the coherence effects we ignore the crosstalk and start from an ideal infinite hologram, recorded by a point source which results in the expression given by Eq. (A25). That relation gives, apart from a quadratic phase factor common to the whole detector plane, the complex amplitude distribution over a single detector pixel due to a single hologram. This relation is reproduced here for convenience:

$$U'' = \sum_{mn} g_{mn} h_{ijmn} \mathcal{G}(-\mathbf{d}^s_{mn}/f)\, \mathrm{sinc}(xp_s/\lambda f, yp_s/\lambda f). \qquad (26)$$

The various parameters are explained in the Appendices (we changed the dummy indices to avoid confusion with the wavenumber $k = 2\pi/\lambda$), and here we just note that the g and h factors are the amplitude transmittances of the SLM pixels during the reconstruction and hologram writing processes, respectively. For our discussion here the important factor in this expression is the linear phase factor

$$\mathcal{G}[-\mathbf{d}^s_{mn}/f] = \exp\left[\frac{jk}{f}(md_s x + nd_s y)\right]. \qquad (27)$$

The power detected by the detector element is given by

$$|U''|^2 = \sum_{mn} \sum_{m'n'} g_{mn} h_{ijmn} g_{m'n'} h_{ijm'n'} \times \mathcal{G}(-\mathbf{d}^s_{mn}/f) \mathcal{G}(\mathbf{d}^s_{m'n'}/f)\, \mathrm{sinc}^2(xp_s/\lambda f, yp_s/\lambda f). \qquad (28)$$

The terms in the double summation are the same for both summations. Therefore, the mixed terms occur twice with the sign of the linear phase inverted. Thus we may write

$$|U''|^2 = \mathrm{sinc}^2\left(\frac{xp_s}{\lambda f}, \frac{yp_s}{\lambda f}\right) \times \left\{\sum_{mn} g^2_{mn} h^2_{ijmn} + \sum_{m<m', n<n'} g_{mn} h_{ijmn} g_{m'n'} h_{ijm'n'} \times \cos\left[k \frac{(\mathbf{d}^s_{mn} - \mathbf{d}^s_{m'n'}) \cdot \boldsymbol{\rho}}{f}\right]\right\}. \qquad (29)$$

This relation gives the power distribution over the detector plane for reconstruction with a single hologram having infinite size. To find the actual detector signal we must integrate over its sensitive area p_d^2. Then the first sum will correspond to Eq. (1) if we put

$$g^2_{kl} \propto a_{kl}; \qquad h^2_{ijkl} \propto t_{ijkl}, \qquad (30)$$

and we are left with the disturbing interference terms of the second summation. Due to the integration, the contribution of these interference terms is quite small except for the smallest phase factors contributed by nearly neighboring pixels. The worst case is a nearest neighbor such as $m = m', n = n' - 1$. For this case we shall have to integrate over an expression of the form

$$\mathrm{sinc}^2(xp_s/\lambda f, yp_s/\lambda f)\left[\alpha + \beta \cos\left(\frac{dd_s}{f} x\right)\right], \qquad (31)$$

where α and β are two constants determined by the input vector and interconnection strengths. Since in the derivation of Eq. (26) we have already extracted the shift operator from the amplitude distribution [Eq. (A22)], we assume here that the detector pixel is centered at ($x = 0, y = 0$), where, according to the above relation, the intensity is much higher than for the incoherent superposition. The amount of this discrepancy, after integration, depends on detector pixel size p_d. If we take this size to cover the region up to the first zero of the sinc function as in Eq. (26), we obtain the first zero of the interference at

$$x_i = \frac{\lambda f}{4d_s}, \qquad (32)$$

which satisfies the relation $x_l < p_d/4$, and we have at least one full oscillation period over the integration length.

The actual net effect will be a redistribution of the power over the detector area without much effect on the total power. As a matter of fact, for this special choice of parameters a larger fraction of the power is concentrated around the detector center, which tends to reduce crosstalk. If we calculate the integrated power over the detector for this case we obtain 25% more power concentrated on the detector surface. However, considering the fact that the relation $d_s > p_s$ is always satisfied, the coherent contribution is much less pronounced. Taking the parameters of the previous section, $d_s = 2p_s$, the difference between coherent superposition and noncoherent superposition is only 0.2%. More distant pixels will contribute interference terms of higher spatial frequency, which tends to reduce the uncertainty to a negligible amount. The uncertainty in the detected value is also increased by the fact that no prior knowledge is possible about the relative magnitudes of the parameters α and β.

An additional error source due to coherent illumination comes from contamination and irregularities in the actual system. For example, a dust particle of cross-sectional area having a fraction η of a pixel area may scatter light of that fraction very unevenly. Thus a fraction η of power may be removed from one pixel and injected into another. A 10-μm particle, common to most laboratory environments, has an area of the order of 0.3% of the SLM pixel area we derived in Sec. IV. If we consider this fractional noise as a coherent amplitude noise, it may amount to ~0.6% of local power uncertainty.

To complete this picture one must also consider the fact that most contributions to the crosstalk error are coherent with the signals, and for accurate analysis Eq. (13) should be modified accordingly.

For the sake of brevity we considered in detail only the coherence effects at the detector plane. We should keep in mind, however, that similar effects occur also during the hologram recording process and at the SLM during reconstruction. Although we assume linear interaction with the SLM, a redistribution of intensities due to coherent superposition may contribute to an increased uncertainty in the interconnection weights. As on the detector plane, where we investigated the power redistribution due to the coherent superposition of the contribution from different SLM pixels, on the SLM plane we have the same effect from the coherent superposition of the contributions from different holograms. However, as mentioned earlier, assuming linear interaction at the SLM this coherent superposition has no appreciable contribution to the error.

Some deteriorating coherence effects may be reduced by adding a random phase mask[24] to the SLM as noted. If the spatial frequency of this phase mask is higher than $1/p_s$, this will increase the IC requirements of the hologram, and information will be lost unless the hologram size is increased. If, however, the phase mask has a constant phase over each pixel this may spread the information more evenly without contributing much to bandwidth requirements. A phase modulation of this kind may also reduce the uncertainties induced by coherent superposition over the various planes, although some precautions should be exercised to avoid unnecessary distortions.

VII. Polarization Effects and SLM Performance

Most SLMs that are available today operate on the polarization of light, and they are designed for near normal incidence. If an SLM is to be employed with light beams having variable angles of incidence, we are faced with two major effects. The first effect concerns the angular dependence of the SLM performance itself, and the second effect is related to the polarization characteristics of nonplanar wavefronts.[28] According to some recent investigation, reflective magnetooptic SLMs may be designed with reduced angular dependence.[29] However, most SLMs available today rely on the transmission of light through a controlled birefringent medium, such as a liquid crystal. This birefringent medium has a thickness d within which the optical path difference for the two polarization components $l(\theta)$ changes with angle. If we assume the birefringent layer to function as a halfwave plate, we may write the relation

$$l(0) = (n_e - n_o)d = (2N + 1)\lambda/2, \tag{33}$$

where $l(0)$ is the optical path difference for normal incidence, N is an integer giving the order of the waveplate, and n_o, n_e are the ordinary and extraordinary indices, respectively. If a light beam is incident on the face of the SLM at an angle of incidence θ, then, taking into account Snell's law, the optical path difference changes to

$$l(\theta) = \frac{(n_e - n_o)d}{\sqrt{1 - \frac{\sin^2\theta}{n^2}}} = \frac{(2N + 1)\lambda}{2\sqrt{1 - \frac{\sin^2\theta}{n^2}}}, \tag{34}$$

where we ignored the splitting of the two polarization components and took n as some average of n_e and n_o. Assuming a relatively small angle, we may write the approximate relation

$$l(\theta) = \frac{(2N + 1)\lambda}{2}\left(1 + \frac{\sin^2\theta}{2n^2}\right). \tag{35}$$

this relation means that we have introduced a phase error of the order of

$$\phi \approx (2N + 1)\pi \frac{\sin^2\theta}{2n^2}. \tag{36}$$

This phase error approximately determines the value of the field component emerging from the SLM medium at the wrong polarization that will contribute a fractional error in the power transmitted by the analyzer of the order of

$$\sin^2\left[(2N + 1)\pi \frac{\sin^2\theta}{2n^2}\right]. \tag{37}$$

Polarization errors of this kind may become quite large for high-order waveplates, but it is rather small for first-order plates ($N = 0$) provided the angle is not large.

Special SLM design for broader acceptance angle may be helpful in reducing this effect. However, even with a reduced angular sensitivity of the SLM itself, polarizers must be employed in conjunction with them, at least as long as they operate on the polarization state of light. Thus we should consider the deterioration of the performance of polarizers with a varying angle of incidence.[28] We evaluate this degradation of performance by first defining a polarization unit vector $\hat{P}$ that describes the polarization orientation of light transmitted by a polarizer for a normally incident light beam. Denoting the propagation direction of any incident beam by a unit vector $\hat{k}$, the transmitted beam will have a polarization orientation along the unit vector,[28]

$$\hat{e} = \frac{\hat{P} - \hat{k}(\hat{k}\cdot\hat{P})}{\sqrt{1-(\hat{k}\cdot\hat{P})^2}} . \tag{38}$$

If we place now an analyzer with its polarization vector $\hat{P}'$, following the polarizer, the transmitted field component $\mathbf{e}_t$ will be the projection of $\hat{e}$ on a similar vector, $\hat{e}'$ determined by $\hat{P}'$ also according to Eq. (38). With an incident beam of unit amplitude, the transmitted amplitude component will be

$$e_t = \hat{e}\cdot\hat{e}' = \frac{\hat{P} - \hat{k}(\hat{k}\cdot\hat{P})}{\sqrt{1-(\hat{k}\cdot\hat{P})^2}} \cdot \frac{\hat{P}' - \hat{k}(\hat{k}\cdot\hat{P}')}{\sqrt{1-(\hat{k}\cdot\hat{P}')^2}} . \tag{39}$$

Evaluating this scalar product for crossed polarizers ($\hat{P} \perp \hat{P}'$) we obtain the fraction of field amplitude that leaks through:

$$|e_t| = \frac{(\hat{P}\cdot\hat{k})(\hat{P}'\cdot\hat{k})}{\sqrt{1-(\hat{P}\cdot\hat{k})^2}\sqrt{1-(\hat{P}'\cdot\hat{k})^2}} . \tag{40}$$

With our definition for the maximum inclination angle [Eq. (5)], one may show that the maximum value of the scalar products in this equation is given by (see Fig. 2)

$$\hat{P}\cdot\hat{k} = \hat{P}'\cdot\hat{k} = \sin\alpha = \frac{\sin\theta_{\max}}{\sqrt{2}} . \tag{41}$$

Considering the amplitude transmission of two crossed polarizers as given in Eq. (40), we may interpret the square of this expression as the angle dependent extinction ratio that attains its worst value for the maximal angle of incidence [Eq. (5)]:

$$|e_t|^2_{\max} = \tan^4\alpha \approx \frac{\sin^4\theta_{\max}}{4} , \tag{42}$$

where the approximation applies for small angles that do not always apply. This error is comparable to first-order ($N = 0$) SLM power polarization error [Eq. (37)] that for relatively small angles may be approximated by

$$\frac{\pi^2 \sin^4\theta_{\max}}{4n^4} . \tag{43}$$

If, for example, we take the large value $\theta_{\max} = 45°$ as for the estimations in Sec. IV, we obtain an extinction ratio of the order of 1/9, even with an ideal SLM combined with ideal polarizers.

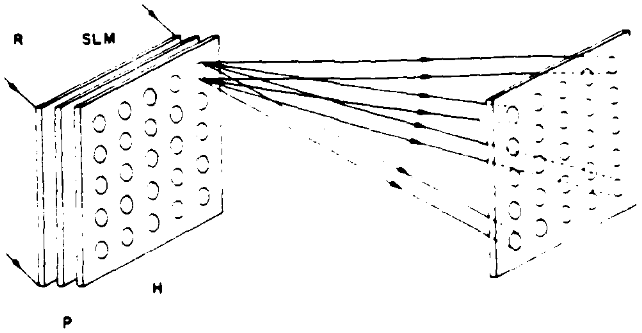


Fig. 5. Modified architecture with SLM S adjacent to the hologram array H. P is an optional high efficiency grating that may be employed for tilting the reference beam.

Similar polarization effects play a role also during the recording of a hologram. While the reference beam has a well-defined polarization the object beam is usually not a uniform plane wave. If we adjust the polarization of the reference beam to fit the polarization of part of the object wavefront, all beam components incident on the hologram at different angles will have a polarization error effecting the reconstruction beam diffraction efficiency into that direction.

VIII. Modified Architectures

The most severe limitation in the original architecture is due to diffraction crosstalk and the angular constraints of the SLM. In a modified system, suggested also in Refs. 30 and 31, the SLM is attached to the hologram array, and it is illuminated by a uniform reference beam (Fig. 5). Thus all the angular variations of light incident on the SLM are eliminated. Furthermore, there is no longer a need for large lenses in the operating system, although they may be needed for the hologram recording stage.

The system is operated here too by introducing the input vector a_{kl} in the SLM. Each element of the hologram array is illuminated by the reference beam through a corresponding pixel in the SLM, thus with a reference beam-intensity proportional to a_{kl}. The klth hologram diffracts light toward the ijth detector in the array with an efficiency t_{klij}; this detector receives from the klth hologram light with power proportional to $t_{klij}a_{kl}$. The overall power detected by this detector element is the sum of all the contributions (again ignoring coherence effects):

$$b_{ij} = \sum_{kl} t_{klij}a_{kl} . \tag{44}$$

This equation is of the same form as Eq. (1), which contains t_{ijkl} rather than $tklij$ and will be identical to it if the new $\|T\|$ matrix is the transpose of the old one.

One possible procedure to record the hologram array is similar to the original architecture and uses the same system. This reinstalls some of the problems discussed earlier but which can be dealt with more efficiently since for every hologram recording, a single source is used.

The main initiative for this architecture was the reduction of the angular constraints on the SLM, and we have already seen that it also eliminates the need for the lenses. It turns out that there are also additional advantages: The polarization effects were also mainly related to the function of the SLM, and, therefore, they are absent here. Furthermore, there is no crosstalk on the SLM plane since it is practically in contact with the hologram. Thus in this architecture the only crosstalk (with an ideal SLM having infinite extinction ratio) is on the detector plane and is determined by the diffraction spot size of the hologram and will satisfy the relation

$$\epsilon_d \approx \left(\frac{\lambda f}{D_d p_h}\right)^2 \tag{45}$$

and also by coherence effects and aberrations.

The pixel size of the SLM is virtually unlimited for this architecture since each pixel serves only as the source of a reconstruction beam for a single hologram. Small SLMs available today may be used with a projection optical system to match its pixel size optically to that of the required hologram aperture.

The most obvious penalty for the benefits of this architecture is the reduced flexibility due to the requirement that there should be a one-to-one correspondence between the individual holograms and SLM pixels. Thus overlapping holograms are no longer allowed, and also the same number of SLM pixels as holograms must be used unless one allows the illumination of several holograms with a single SLM pixel or vice versa. This state of affairs is useful for many applications where the overall N^4 interconnections are not required.

Coherent superposition at the detector plane still takes place as with the original architecture, but here the superposition is from different holograms and not from different SLM pixels. To overcome this problem an improved version of this architecture will be eventually possible with the development of large laser diode arrays that may be able to replace the SLM. The lasers in the array will be individually modulated to represent the input vector.

This modification will lead to an appreciable reduction in coherent noise since each hologram is illuminated by a separate laser and each detector element receives a single contribution from each hologram. Now the superposed beams on each detector element originate from different lasers and may be combined incoherently (see also Appendix A).

The diode array configuration will be superior to the various SLM configurations in speed, dynamic range, and SNR. The only obvious problem is that such arrays are not available yet; however, present research in this area should provide the needed devices in the relatively near future.

IX. Illumination Power Requirements

To estimate the laser power requirements we denote the minimum detectable power by w_o. If we allow n interconnection weight levels, we would ideally like to have a maximum power available in each channel at the detector plane equal to $n \times w_o$. Each hologram in the hologram array diffracts light into N_s^2 channels, and we have N_h^2 such holograms (and, of course, also $N_d^2 = N_h^2$ detectors). Thus the maximum total diffracted power we need at the detector plane will be given by

$$W_d = N_h^2 N_s^2 n w_o. \tag{46}$$

Denoting by η an overall efficiency of the reconstruction process (including hologram diffraction efficiency, useful hologram area, SLM, and detector cross-sections as well as other losses in the system), we arrive at a laser power of

$$W_l = \frac{N_h^2 N_s^2 n w_o}{\eta}. \tag{47}$$

This power requirement may be very high although affordable. It turns out, however, that we really do not need such high power levels since this calculation assumes a detection dynamic range of $N_h^2 N_s^2 n$, which may become also too high for any reasonable detector. Furthermore, if we take into account the unavoidable noise level given by Eq. (25), we may conclude that there is no sense in requiring a detection level which is lower than this noise. Thus, assuming that in Eq. (25) we always have $\epsilon_d > 1$ for a realistic system, we may require a detection limit of only ϵ_d and decision levels also not closer than this value. Basing this realistic approach on the considerations that lead to Eq. (25) we may derive a new value for the laser power: We set the *minimum detectable power equal to the maximum* crosstalk noise,

$$w_o = (I\epsilon_d)_{\max} = MN_s^2 I\epsilon_f, \tag{48}$$

where I is the power received by the detector from a fully weighted channel. The maximum power received by a single detector will be when all channels are addressed to it, that is, $N_s^2 I$. Therefore, we may assume a worst case number of decision levels to be

$$\frac{N_s^2 I}{(I\epsilon_d)_{\max}} = \frac{1}{M\epsilon_f}. \tag{49}$$

The number of decision levels multiplied by the detector sensitivity w_o gives the maximum required power on the detector. For N_d^2 such detectors and taking into account the overall efficiency, we obtain the total laser power

$$W_l = \frac{N_d^2 w_o}{M\epsilon_f \eta}. \tag{50}$$

X. Design Considerations

Design parameters for an actual system are very strongly application dependent, and before attempting any design procedure one should answer several questions: What is the dimension of the vectors to be processed? What is the minimum acceptable number of interconnection weight levels? What is the minimum number of detection levels? What is an acceptable error? Some of the answers to these questions may turn out to be incompatible due to the limitations derived in this paper, and compromises should be made. For example, we have seen that with an acceptance angle of 45° the extinction ratio of an SLM system will be worse than 1:9, and crosstalk may be at least a few percent. This limits the number of weight levels to ~10. A smaller more realistic angle may provide a larger number of weight levels, but then the number of matrix elements is greatly reduced.

As a case study we attempt to design a neural network with input and output vectors of 256 × 256 elements. Several neural network algorithms use a thresholded two-level output but with interconnection weights requiring a dynamic range as large as possible.

Hologram arrays of this size have already been implemented,[10,11] and what remains for us here is to determine their limitations. Taking in Eq. (11) $l \approx 1$ μm and $g_s = g_h$ we obtain $p_h \geq 256$ μm as was actually implemented (with the lower limit) in the above references. To accommodate also the off-axis carrier for the hologram recording we need at least another factor of 2 that leads to $p_h = 500$ μm.

If we use rectangular pixels, the various sizes will be determined by the allowable crosstalk using calculations like those for the plots of Fig. 4. Taking an F/no. = 1 leads to $\theta_{\max} = 30°$. Considering the architecture of Fig. 1 with an SLM layout similar to Fig. 3 we may attempt to choose a value $d_s/p_s = 2$ with p_s matching the central lobe of the hologram diffraction pattern. Using the calculated results of Sec. VI, we obtain a fractional crosstalk from a single neighboring pixel of the order of $\epsilon_f \approx 0.75\%$ and $M\epsilon_f \approx 4\%$. The polarization errors from relations (42) and (43) will be of the order of 4.5% assuming $n \approx 1.5$. The total uncertainty in the weights is thus close to 10%, and there is not much sense in requiring more than ten weight levels.

Returning now to the analysis of Sec. IV we may modify relation (8) to

$$\frac{d_s}{\lambda} \geq \frac{2\sqrt{2}}{\sin 60°}\,\frac{H}{p_h} = 3.27\,\frac{H}{p_h}\,. \qquad (51)$$

For 256 pixels with nonoverlapping holograms we need $H/p_h = 256$ leading to $d_s/\lambda = 836$. With $\lambda = 0.5$ μm we end up with $d_s = 418$ μm. This value is larger than available in most SLMs, and we may overcome this difficulty by partially overlapping holograms: A value of $p_h = 2d_h$ bring us into a practical domain.

To consider the crosstalk over the detector plane we assume a detector layout similar to Fig. 3, keeping the condition of Eq. (51) with d_d replacing d_s and p_s replacing p_h. In these conditions we obtain a maximum estimated error given by Eq. (25) to be $\sim 256^2 \times 4\% I \approx 2600 I$ with I denoting the full weighted single-channel interconnection power. At first sight this appears to be a formidable error, but it still constitutes just 4% of the maximum power to be detected, and we may use twenty-five decision levels.

If we convert to the modified architectures of Fig. 5 we may forget about the crosstalk over the SLM and its angular dependence, although special precautions should be taken during hologram recording. By Eq. (51) we can replace d_s by d_d and note that the detector error remains essentially the same.

The power requirements can be estimated from Eq. (49). Taking the estimated $\epsilon_d = M\epsilon_f \approx 4\%$ and assuming an overall light efficiency $\eta = 0.1$, we shall need a laser power of $1.64 \times 10^7\ w_o$. With high sensitivity detectors $w_o = 10^{-9}$ W we need a laser power of ~16 mW, which implies that laser power is not a problem, and one may employ detectors of lower sensitivity when used in conjunction with higher power lasers.

The required detector is also available. If we take the obvious detector size value $d_d = d_s = 450$ μm leading to $D = 450 \times 256 = 11.52$ cm we cannot use the commercial CCD arrays, but we can use solar cells, vidicons, or image dissectors for detection and readout.

To conclude this section we may safely state that $256^4 = 4.3 \times 10^9$ interconnections between $256^2 = 6.55 \times 10^4$ channels with at least 10 weight levels, and twenty-five output levels are possible with presently available devices. Assuming a TV rate of 30 frames/s we may perform $30 \times 256^4 \approx 1.3 \times 10^{11}$ interconnections/s. This number may be increased at least by 1 order of magnitude at the expense of the number of weight levels (or just the decision levels in the modified architecture). With progress in the technology of SLMs we may expect one more order of magnitude, and the limit is still greater once large and fast laser diode arrays are available.

Assuming a conservative number of 3-bit weight levels leads to a total of IC that is stored in the hologram array of $3 \times 256^4 = 1.3 \times 10^{10}$ bits.

XI. Conclusions

We have shown in this work that it is possible to implement with existing devices a holographic weighted interconnection network with 256 × 256 channels. The various constraints have been analyzed in detail, and proper design parameters were evaluated. Before implementing a working system, however, a few important remarks should be observed:

In the estimation of constraints and errors we based most of the calculations on worst case conditions. Therefore, we may expect actual performance levels to be much better than stated here because worst case parameter values are seldom encountered. Furthermore, the appearance of all the worst case parameters together will be extremely rare.

We have described two architectures, and at this point it is difficult to say which is preferred since preference appears to be application dependent. While in the modified architecture one may definitely use a larger number of weight levels, in the first architecture more channels may be incorporated due to the possibility of overlapping holograms.

In many applications one does not need all the possible interconnections as in the human brain where only a very small fraction of the total number of neurons are actually interconnected. For cases like this one may arrange the channels in the array so that most of the interconnections are made among nearby channels from the constraints point of view. Such a sectioning of the array will allow a significant increase in the overall array size since our constraints must be kept only within a single section.

Finally, we should point out that in this application the holographic reconstruction produces the complex conjugate of the writing wavefront. Therefore, while propagating through the optical system more phase distortions are compensated if they are the same as during hologram recording. This means that the optical components do not have to be of the highest quality, although they must perform adequately well for the hologram recording.

Appendix A: Paraxial System Analysis

The purpose of this Appendix is evaluation of the performance of an ideal system to realize the limitations introduced by fundamental physical processes. We calculate the system transfer characteristics using paraxial approximation and the operator notation which is summarized in Appendix B. We also assume ideal SLM performance and hologram reconstruction. To simplify the expressions we ignore constant phase and amplitude factors that will affect the signal and noise in the same way; i.e., they are not important for this discussion. In any case, if these factors are needed, they can be reinstalled by using simple physical considerations.

We start these Appendices from the hologram recording stage, which is the same for the original architecture as well as for the modified architectures and then discuss separately the three configurations.

A. Hologram Recording

For a discussion of the hologram recording stage we return to Fig. 1 and the notations used in the main text. We introduce a light source with complex amplitude $u(x,y)$ at the location of the ijth detector. Without losing generality we may simplify the notation by assuming $f_1 = f_2 = f$. The complex amplitude U_s incident on the SLM can be expressed in operator form by the relation

$$U^{s}_{ij} = \mathcal{Q}[-1/f]\mathcal{R}[f]\mathcal{S}[id_d\hat{x} + jd_d\hat{y}]u(x,y), \tag{A1}$$

where we introduced the shift operator $\mathcal{S}$ [Eq. (B5)] to represent the position of the source and denoted by $\hat{x}$ and $\hat{y}$ the unit vectors along x and y, respectively. The input complex amplitude is operated on by $\mathcal{R}[f]$, the free space propagation operator (FPO) [Eqs. (B7)–(B9)] through a distance f and is finally multiplied by the quadratic phase factor $\mathcal{Q}[-1/f]$ [Eq. (B1)], which is the transfer operator of an ideal thin lens [Eq. (B6)]. This distribution is multiplied by the SLM transfer function specified for the ijth recording,

$$h_{ij}(x,y) = \sum_{kl} h_{ijkl}\mathcal{S}[kd_s\hat{x} + ld_s\hat{y}]\ \mathrm{rect}(x/p_s,y/p_s), \tag{A2}$$

where we again employed the shift operator to place a rectangular window function at the proper position of each pixel. This SLM transfer function is again transformed by the second lens propagated a distance f and finally recorded on the hologram at position ij as

$$U_{ij} = \mathcal{R}[f]\mathcal{Q}[-1/f]h_{ij}(x,y)U^{s}_{ij}. \tag{A3}$$

Returning to Eq. (A1) we substitute Eq. (B18) for the FPO and use Eq. (B8) to obtain

$$U^{s}_{ij} = \nu[1/\lambda f]\mathcal{F}\mathcal{Q}[1/f]\mathcal{S}[\mathbf{d}^{d}_{ij}]u(x,y), \tag{A4}$$

where we defined

$$\mathbf{d}^{d}_{ij} = id_d\hat{x} + jd_d\hat{y}. \tag{A5}$$

Using Eq. (B13) to commute the shift operator with the quadratic phase we obtain

$$U^{s}_{ij} = \nu[1/\lambda f]\mathcal{F}\mathcal{S}[\mathbf{d}^{d}_{ij}]\mathcal{G}[\mathbf{d}^{d}_{ij}/f]\mathcal{Q}[1/f]u(x,y). \tag{A6}$$

The operation of the Fourier transform (FT) operator is evaluated by using Eqs. (B15) and (B16) to yield

$$U^{s}_{ij} = \nu[1/\lambda f]\mathcal{G}[-\lambda\mathbf{d}^{d}_{ij}]\mathcal{S}[\mathbf{d}^{d}_{ij}/\lambda f]\mathcal{F}\mathcal{Q}[1/f]u(x,y). \tag{A7}$$

Operating with the scaling operator we obtain

$$U^{s}_{ij} = \mathcal{G}[-\mathbf{d}^{d}_{ij}/f]\mathcal{S}[\mathbf{d}^{d}_{ij}]\nu[1/\lambda f]\mathcal{F}\mathcal{Q}[1/f]u(x,y). \tag{A8}$$

If the source is an ideal point source, i.e., $u(x,y) = \delta(x,y)$, the quadratic phase factor is eliminated by the sifting theorem, and, recalling that the FT of a delta function is unity, we obtain a displaced plane wave traveling at an angle defined by the position of the point source:

$$U^{s}_{ij} = \mathcal{G}[-\mathbf{d}^{d}_{ij}/f]\mathcal{S}[\mathbf{d}^{d}_{ij}]. \tag{A9}$$

Naturally, the displacement of a plane wave has no meaning. Thus we are left with a linear phase factor as it should be. The finite extent of the source introduces an apodizing factor to the illumination of the SLM plane. Thus the illumination of the SLM will not be uniform, and this nonuniformity will be shifted according to source location. Usually the source will be approximately a Gaussian source, and the complex amplitude distribution over the SLM plane can be predetermined and partially compensated for. Keeping this in mind we proceed with the assumption of an ideal point source.

Substitution into Eq. (A3) together with Eq. (B18) leads to

$$U_{ij} = \mathcal{Q}[1/f]\nu[1/\lambda f]\mathcal{F}h_{ij}(x,y)\mathcal{G}[-\mathbf{d}^{d}_{ij}/f]. \tag{A10}$$

Performing the FT operation we obtain

$$U_{ij} = \mathcal{Q}[1/f]\nu[1/\lambda f]\mathcal{S}[-\mathbf{d}^{d}_{ij}/\lambda f]H_{ij} = \mathcal{Q}[1/f]\mathcal{S}[-\mathbf{d}^{d}_{ij}]\nu[1/\lambda f]H_{ij}, \tag{A11}$$

where H_{ij} is the FT of h_{ij}. If we need to reinstall the source distribution we should substitute here and in the following:

$$\nu[1/\lambda f]H_{ij} \rightarrow (\nu[1/f]H_{ij}) * [\mathcal{G}[-d/f]\nu[-1]\mathcal{Q}[1/f]u(x,y)], \tag{A12}$$

where we used some operator algebra and the * denote convolution. The complex amplitude of Eq. (A11) is recorded as the ijth hologram throug an aperture or more generally through some window function $w(x,y)$ that apart from a limiting aperture may also include some apodization function. In most cases, however, this window function will be of the form

$$w(x,y) = \mathrm{rect}\left(\frac{x}{p_h},\frac{y}{p_h}\right). \tag{A13}$$

It should be noted here that a reinstallation of the finite source size would contribute a convolution over the hologram plane tending to spread the flux and reduce the information content of the system.

B. Reconstruction and Operation—Original Architecture

The hologram array is reconstructed with the conjugate of the reference beam. Thus, for each hologram, we reconstruct the complex conjugate of an expression like Eq. (A11) multiplied by the window function $w(x,y)$. The reconstructed beam is propagated through a distance f, multiplied by the lens transfer function and then by a new function $g(x,y)$, written on the SLM. This new function is the input vector to the processing system.

$$g(x,y) = \sum_{mn} g_{mn}\mathcal{S}[md_s\hat{x} + nd_s\hat{y}]\ \mathrm{rect}(x/p_s,y/p_s). \tag{A14}$$

After multiplication by $g(x,y)$ the complex amplitude is transformed by the second lens and propagated a distance f to the detector plane that will receive the complex amplitude from hologram ij,

$$\begin{aligned} U^{d}_{ij} = {} & \mathcal{R}[f]Q[-1/f]g(x,y)Q[-1/f]\mathcal{R}[f] \\ & \times \{\mathcal{S}[\mathbf{d}^{h}_{ij}]w(x,y)\}\mathcal{S}[-d^{d}_{ij}]\nu[1/\lambda f]H^{*}_{ij}, \end{aligned} \tag{A15}$$

where the complex conjugate of Eq. (A11) has been substituted and modulated by the properly positioned window function. The overall field distribution on the detector plane will be the coherent summation from all the holograms,

$$U_d = \sum_{ij} U^{d}_{ij}. \tag{A16}$$

Using Eq. (B18) and then Eq. (B8) in Eq. (A15) leads to

$$U^{d}_{ij} = Q[1/f]\nu[1/\lambda f]\mathcal{F}g(x,y)\nu[1/\lambda f]\mathcal{F}\mathcal{S}[-\mathbf{d}^{d}_{ij}]w(x,y)\nu[1/\lambda f]H^{*}_{ij}, \tag{A17}$$

where we took into account that the hologram plane is the inverted image of the detector plane with unit magnification, that is,

$$\mathcal{S}[\mathbf{d}^{h}_{ij}] = \mathcal{S}[-\mathbf{d}^{d}_{ij}] \tag{A18}$$

and employed the definition of the shift operator [Eq. (B6)]. Commuting the shift operator with the FT operator using Eqs. (B14) and (B16), we obtain

$$U^{d}_{ij} = Q[1/f]\nu[1/\lambda f]\mathcal{F}g(x,y)\nu[1/\lambda f]\mathcal{G}[\lambda\mathbf{d}^{d}_{ij}]\mathcal{F}w(x,y)\nu[1/\lambda f]H^{*}_{ij}. \tag{A19}$$

Moving now the linear phase factor more to the left we may write

$$U^{d}_{ij} = Q[1/f]\mathcal{S}[\mathbf{d}^{d}_{ij}]\nu[1/\lambda f]\mathcal{F}g(x,y)\nu[1/\lambda f]\mathcal{F}w(x,y)\nu[1/\lambda f]H^{*}_{ij}. \tag{A20}$$

Now we may translate the right-hand side scaling operator to the left and combine it with the middle one to obtain

$$U^{d}_{ij} = Q[1/f]\mathcal{S}[\mathbf{d}^{d}_{ij}]\nu[1/\lambda f]\mathcal{F}g(x,y)\mathcal{F}w(\lambda fx,\lambda fy)H^{*}_{ij}. \tag{A21}$$

The overall process generated an extra quadratic phase factor that is not important for the detection while the shift operator places the center of the distribution at the proper pixel on the detector. The signal to be detected and the amount of crosstalk to be expected are determined by the rest of the expression. To analyze the various contributions of the functions involved let us start with an unlimited hologram recording, that is, $w(x,y) = 1$. To shorten the equations we define a new amplitude U' by the relation

$$U^{d}_{ij} = Q[1/f]\mathcal{S}[\mathbf{d}^{d}_{ij}]U'. \tag{A22}$$

Putting $w(x,y) = 1$ and performing the right-hand side FT operation in Eq. (A21) we obtain

$$U' = \nu[1/\lambda f]\mathcal{F}g(x,y)h^{*}(-x,-y), \tag{A23}$$

which essentially is the cross-correlation of the FT of $g(x,y)$ and $h(x,y)$. With our special functions [Eqs. (A2) and (A14)], however, it is easier to substitute them right away and observe that they are real and the rect function is symmetric. Both functions are composed of the same rect function with terms shifted to various positions. Since the centers are spaced at distances that are multiples of d_s that are never smaller than the extent of the function itself p_s, the only nonzero terms in the product of the two sums are those having $kl = mn$. Furthermore, the square of the rect function is also the function itself; thus we may write

$$U' = \nu[1/\lambda f]\mathcal{F}\sum_{kl} g_{kl}h_{ijkl}\mathcal{S}[\mathbf{d}^{s}_{kl}]\ \mathrm{rect}(x/p_s,y/p_s), \tag{A24}$$

where we introduced a position vector similar to Eq. (A5). Performing the Fourier transformations and the scaling operations lead to the final expression

$$U' = \sum_{kl} g_{kl}h_{ijkl}\mathcal{G}[-\mathbf{d}^{s}_{kl}/f]\ \mathrm{sinc}(xp_s/\lambda f,yp_s/\lambda f). \tag{A25}$$

In a real situation we should reinstall the source distribution $u(x,y)$, which should be convolved with the sinc function to obtain the final distribution.

Some of the consequences of relation (A25) are discussed further in the main text while here we continue by returning to the complete expression that contains the window function but assuming a source distribution much smaller than the sinc distribution. If we do this, instead of Eq. (A23) we have

$$U' = \nu[1/\lambda f]\mathcal{F}g(x,y)\mathcal{F}w(\lambda fx,\lambda fy)H^{*}_{ij}. \tag{A26}$$

If we consider only the right-hand side FT we are essentially on the SLM plane, and we see that there the distribution is given by a convolution of the FT of the window function with the interconnection function h. Thus this distribution is widened contributing to the crosstalk over the SLM plane. However, in the final distribution, that may be written in the form

$$U' = \nu[1/\lambda f]\{\mathcal{F}g(x,y)\} * \{w(x,y)\nu[1/\lambda f]H^{*}_{ij}\}, \tag{A27}$$

where $*$ denotes convolution. The window function actually reduces the crosstalk over the detector plane since the width of H_{ij} is cut by the window function before performing the convolution operation with the FT of the input vector g.

C. Reconstruction and Operation-Modified Architectures

In the modified architectures (Fig. 5) the hologram recording process is similar to that of the original configuration. Thus we may start here also from the reconstruction where now each hologram is multiplied directly by its input vector element. The complex amplitude is now propagated a distance f, multiplied by the lens transfer function, and detected on the detector plane. The lens has no effect on our results here, but it is useful in correcting for a quadratic phase factor to reduce the variation in the angle of incidence over the detector plane. Denoting the complex amplitude of the light transmitted by the ijth element of the input vector by g_{ij}, it reconstructs the ijth hologram to generate the distribution over the detector plane given by

$$\begin{aligned} U_{ij}^{d} &= Q[-1/f]\mathcal{R}[f]g_{ij} \\ &\times \{\mathcal{S}[-\mathbf{d}_{ij}^{d}]w(x,y)\}Q[-1/f]\mathcal{S}[-\mathbf{d}_{ij}^{d}]\nu[1/\lambda f]H_{ij}^{*}, \end{aligned} \tag{A28}$$

where we introduced again the shifted window function that delineates the hologram (and now also the SLM pixel) size. Substituting Eq. (B18) for the FPO, using Eq. (B8), and taking into account that g_{ij} is just a constant and $\{\mathcal{S}[-\mathbf{d}_{ij}^{d}]w(x,y)\}$ is a scalar function, we obtain

$$U_{ij}^{d} = g_{ij}\nu[-1/\lambda f]\mathcal{F}\mathcal{S}[-\mathbf{d}_{ij}^{d}]w(x,y)\nu[1/\lambda f]H_{ij}^{*}. \tag{A29}$$

Performing the scaling and FT operations leads to

$$U_{ij}^{d} = g_{ij}\mathcal{G}[-\mathbf{d}_{ij}^{d}/f]\{[\nu[1/\lambda f]\mathcal{F}w(x,y)] * h_{ij}^{*}(-x,-y)\}. \tag{A30}$$

Taking into account that h is a real function, composed of symmetric terms (the rect functions), we reconstructed the original h function multiplied by the vector element g_{ij}, but each pixel distribution is now widened due to the convolution with the FT of the window function. This convolution will cause a crosstalk by extending the distribution from each detector pixel into adjacent pixels. To obtain the complete distribution over the detector plane one must sum all the contributions coherently,

$$U^{d} = \sum_{ij} U_{ij}^{d}. \tag{A31}$$

If we substitute Eq. (A2) we obtain a relation similar to Eq. (A25), but this time the linear phase factors originate from the hologram position. The advantage of the laser diode array architecture is that these phase factors are canceled due to the incoherent superposition,

$$|U^{d}|^{2} = \sum_{ij} |U_{ij}^{d}|^{2}. \tag{A32}$$

Appendix B: Summary of Operator Algebra

In this Appendix we summarize the definitions and relevant relations of the operator algebra. For simplicity we shall ignore all constant factors (phase and amplitude) that are irrelevant for the discussions in this paper since we are interested in the complex amplitude distributions and not their absolute magnitudes that can be estimated from simple considerations.

Assuming for all operations a general complex function $f(x,y)$, we define the basic operators as follows. Quadratic phase factor:

$$Q[a] = \exp\left(\frac{jk}{2}a\rho^{2}\right), \tag{B1}$$

with $k = 2\pi/\lambda$, and ρ denoting the transversal coordinate

$$\boldsymbol{\rho} = x\hat{x} + y\hat{y}; \qquad \rho = |\boldsymbol{\rho}|. \tag{B2}$$

Linear phase factor:

$$\mathcal{G}[\mathbf{s}] = \exp(jk\mathbf{s}\cdot\boldsymbol{\rho}). \tag{B3}$$

A scaling operator $\nu[a]$ is defined by the relationship

$$\nu[a]f(x,y) = f(ax,ay)\nu[a], \tag{B4}$$

and the Fourier transform operator is defined by the integral

$$\mathcal{F}f(x,y) = \int f(x',y')\exp[2\pi j(xx' + yy')dx'dy']. \tag{B5}$$

The shift operator is defined by the equation

$$\mathcal{S}[\mathbf{m}]f(x,y) = f(x - m_{x},y - m_{y})\mathcal{S}[\mathbf{m}]. \tag{B6}$$

The transfer operator of an ideal thin lens of focal length f is

$$\mathcal{L}[f] = Q[-1/f]. \tag{B7}$$

Some basic relations are evident from the definitions of the basic operators:

$$Q[a]Q[b] = Q[a + b], \tag{B8}$$

$$\nu[a]\nu[b] = \nu[ab], \tag{B9}$$

$$\nu[a]Q[b] = Q[a^{2}b]\nu[a], \tag{B10}$$

$$\nu[b]\mathcal{S}[\mathbf{m}] = \mathcal{S}[\mathbf{m}/b]\nu[b], \tag{B11}$$

$$\nu[b]\mathcal{G}[\mathbf{m}] = \mathcal{G}[\mathbf{m}b]\nu[b], \tag{B12}$$

$$Q[a]\mathcal{S}[\mathbf{m}] = \mathcal{S}[\mathbf{m}]\mathcal{G}[a\mathbf{m}]Q[a]. \tag{B13}$$

As stated above, constant factors have been ignored in some of these equations and also in the following. Using Fourier analysis we can show that

$$\nu[b]\mathcal{F} = \mathcal{F}\nu[1/b], \tag{B14}$$

$$\mathcal{F}\mathcal{G}[\mathbf{s}] = \mathcal{S}[\mathbf{s}/\lambda]\mathcal{F}, \tag{B15}$$

$$\mathcal{F}\mathcal{S}[\mathbf{m}] = \mathcal{G}[-\lambda\mathbf{m}]\mathcal{F}. \tag{B16}$$

Free space propagation, i.e., the Fresnel-Kirchhoff integral, is described by the FPO, which can be expressed in various ways by the basic operators:

$$\mathcal{R}[d] = \mathcal{F}^{-1}Q[-\lambda^{2}d]\mathcal{F} = \mathcal{F}Q[-\lambda^{2}d]\mathcal{F}^{-1}, \tag{B17}$$

where d is the propagation distance. Another useful expression is

$$\mathcal{R}[d] = Q[1/d]\nu[1/\lambda d]\mathcal{F}Q[1/d], \tag{B18}$$

and for large distances an asymptotic expression may be also employed:

$$\lim_{d\to\infty}\mathcal{R}[d] = \lim_{d\to\infty}\nu[1/\lambda d]\mathcal{F}. \tag{B19}$$

The FPO satisfies the cascading property

$$\mathcal{R}[a]\mathcal{R}[b] = \mathcal{R}[a + b]. \tag{B20}$$

A simple optical system containing a single lens may satisfy the Fourier transforming condition:

$$\mathcal{R}[f]Q[-1/f]\mathcal{R}[d] = Q\left[\frac{1}{f}\left(1-\frac{d}{f}\right)\right]\nu\left[\frac{1}{\lambda f}\right]\mathcal{F} \tag{B21}$$

that becomes exact for $d = f$. Alternatively, the imaging condition,

$$1/a + 1/b = 1/f, \tag{B22}$$

yields

$$\mathcal{R}[a]Q[-1/f]\mathcal{R}[b] = Q\left[\frac{1}{b}\left(1+\frac{a}{b}\right)\right]\nu[-a/b]. \tag{B23}$$

With the basic relations we can also show that

$$\mathcal{R}[d]Q[1/q] = Q[1/(d+q)]\nu[1/(1+d/q)]\mathcal{R}[(1/d+1/q)^{-1}], \tag{B24}$$

$$\nu[b]\mathcal{R}[d] = \mathcal{R}[d/b^2]\nu[b]. \tag{B25}$$

It is a pleasure to thank Y. Fainman, A. Choudry, and R. A. Chipman for useful discussions and Ho-In Jeon for helping with numerical calculations. This work was partially supported by the Office of Naval Research under grant N00014-86-K-0591.

References

1. J. W. Goodman, F. I. Leonberger, S. Kung, and R. A. Athale, "Optical Interconnections for VLSI Systems," Proc. IEEE **72**, 850 (1984).
2. R. K. Kostuk, J. W. Goodman, and L. Hesselink, "Design Considerations for Holographic Optical Interconnects," Appl. Opt. **26**, 3947 (1987).
3. E. Marom and N. Konforti, "Dynamic Optical Interconnections," Opt. Lett. **12**, 539 (1987).
4. Ho-In Jeon and A. A. Sawchuk, "Optical Crossbar Interconnections Using Variable Grating Mode Devices," Appl. Opt. **26**, 261 (1987).
5. J. Shamir and H. J. Caulfield, "High-Efficiency Rapidly Programmable Optical Interconnections," Appl. Opt. **26**, 1032 (1987).
6. K. M. Johnson, M. R. Surette, and J. Shamir, "Optical Interconnection Network Using Polarization Based Ferroelectric Liquid Crystal Gates," Appl. Opt. **27**, 1727 (1988).
7. H. J. Caulfield, "Parallel N^4 Weighted Optical Interconnections," Appl. Opt. **26**, 4039 (1987).
8. K. Wagner and D. Psaltis, "Multilayer Optical Learning Networks," Appl. Opt. **26**, 5061 (1987).
9. J. Kinser, H. J. Caulfield, and J. Shamir, "A Design for a Massive All-Optical Bidirectional Associative Memory: The Big BAM," Proc. Soc. Photo-Opt. Instrum. Eng. **881**, 269 (1988).
10. H. J. White and W. A. Wright, "Holographic Implementation of a Hopfield Model with Discrete Weightings," Appl. Opt. **27**, 331 (1988).
11. P. A. Ambs, Y. Fainman, S. Esner, and S. H. Lee, "Holographic Optical Elements for SLM Defect Removal and for Optical Interconnects," Proc. Soc. Photo-Opt. Instrum. Eng. **883**, paper 31 (1988).
12. P. Ambs, Y. Fainman, S. H. Lee, and J. Gresser, "Computerized Design and Generation of Space Variant Holographic Filter," Proc. Soc. Photo-Opt. Instrum. Eng. **884**, 62 (1988).
13. J.-S. Jang, S.-W. Jung, S.-Y. Lee, and S.-Y. Shin, "Optical Implementation of the Hopfield Model for Two-Dimensional Associative Memory," Opt. Lett. **13**, 248 (1988).
14. N. H. Farhat, "Architectures for Optoelectronic Analogs of Self-Organizing Neural Network," Opt. Lett. **12**, 448 (1987).
15. N. H. Farhat, "Optoelectronic Analogs of Self-Programming Neural Nets: Architecture and Methodologies for Implementing Fast Stochastic Learning by Simulated Annealing," Appl. Opt. **26**, 5093 (1987).
16. K. Fukushima, "Neocognitron: A Hierarchical Neural Network Capable of Visual Pattern Recognition," Neural Networks **1**, 119 (1988).
17. T. Kohonen, "An Introduction to Neural Computing," Neural Networks **1**, 3 (1988).
18. J. Ghosh and K. Hwang, "Optically Connected Multiprocessors for Simulating Artificial Neural Networks," Proc. Soc. Photo-Opt. Instrum. Eng. **882**, 2 (1988).
19. S. C. Gustafson and G. R. Little, "Optical Neural Classification for Binary Patterns," Proc. Soc. Photo-Opt. Instrum. Eng. **882**, 83 (1988).
20. D. Brady, X.-G. Gu, and D. Psaltis, "Photorefractive Crystal in Optical Neural Computers," Proc. Soc. Photo-Opt. Instrum. Eng. **882**, 132 (1988).
21. H. Wagner and R. E. Feinleib, "Competitive Optoelectronic Learning Networks," Proc. Soc. Photo-Opt. Instrum. Eng. **882**, 162 (1988).
22. H. J. White, N. B. Aldridge, and I. Lindsay, "Digital and Analog Holographic Associative Memories," Opt. Eng. **27**, 30 (1988).
23. N. Duklias and J. Shamir, "Relation between Object Position and Autocorrelation Spots in the VanderLugt Filtering Process. 2: Influence of the Volume Nature of the Photographic Emulsion," Appl. Opt. **5**, 78 (1973).
24. Y. Tsunoda and Y. Takeda, "High Density Image-Storage Holograms by a Random Phase Sampling Method," Appl. Opt. **13**, 2046 (1974).
25. M. Nazarathy and J. Shamir, "Fourier Optics Described by Operator Algebra," J. Opt. Soc. Am. **70**, 150 (1980).
26. M. Nazarathy and J. Shamir, "Holography Described by Operator Algebra," J. Opt. Soc. Am. **71**, 529 (1981).
27. M. Nazarathy and J. Shamir, "Wavelength Variation in Fourier Optics and Holography Described by Operator Algebra," Isr. J. Technol. **18**, 224 (1980).
28. Y. Fainman and J. Shamir, "Polarization of Nonplanar Wave Fronts," Appl. Opt. **23**, 3188 (1984).
29. M. N. Deeter and D. Sarid, "Effects of Incidence Angle on Readout in Magnetooptic Storage Media," Appl. Opt. **27**, 713 (1988).
30. R. Clark, C. Hester, and P. Lindberg, "Mapping Sequential Processing Algorithms onto Parallel Distributed Processing Architectures," Proc. Soc. Photo-Opt. Instrum. Eng. **880**, paper 5 (1988).
31. J.-S. Jang, S.-W. Jung, S.-Y. Lee, and S.-Y. Shin, "Optical Implementation of the Hopfield Model for Two-Dimensional Associative Memory," Opt. Lett. **13**, 248 (1988).

APPENDIX C

APPLICATIONS PAPERS

One of the early suggested applications of massive parallelism was to cellular array processing (Opt. Eng. 25, 825). This applies as well to symbolic substitution as practiced by Bell Laboratories (Alan Huang) and to simple associative predictors (Opt. Eng. 25, 1179).

Perhaps the most important SDI application is to massive parallel optical data base management (SPIE 938, 52 and Appl. Opt. 29, Vol. 2, 195). This is the fastest way to search gigabit files.

Of course, these holographic memories can store templates for pattern recognition (SPIE 754, 74) or switching patterns for binary optical devices (SPIE 769, 101) or generalized mapping operations (SPIE 881, 56). The concept of stacked holograms for this purpose also has some promise (SPIE 883, 203).

Systolic optical cellular array processors

H. John Caulfield, FELLOW SPIE
The Center for Applied Optics
The University of Alabama in Huntsville
Huntsville, Alabama 35899

Abstract. Using space-variant pattern recognition of up to 256 3 × 3 patterns of 1's and 0's in parallel and inserting image information sequentially in a well-defined pattern, we can construct an optical systolic cellular array processor for 3 × 3 neighborhoods that produces output points at one-third the rate at which points are input. This allows reprogrammable preprocessing of data input.

Subject terms: Space Station optics; optical processing; cellular array processors; optical computing; systolic arrays.

Optical Engineering 25(7), 825-827 (July 1986).

CONTENTS

1. INTRODUCTION

One of the announced goals of the Space Station program is to include robots for operation and repair in the space environment. While not yet R2D2 clones, these robots must perform tasks well beyond the current state of the art. Typical of such tasks is the tracking (six kinematic parameters per object) of multiple (say, 0 to M) objects from a known set of N ≥ M possible objects. The background clutter is unpredictable. Occultations are probable. Lighting will be nonuniform. Response times must be very fast, say, TV frame time.

Many "tricks" must be applied to make this happen. The Ames Research Center is working on intelligent optical pattern recognition and optical control processing.[*] The Jet Propulsion Laboratory is developing rapid coherent optical data base search methods.[†] The Johnson Space Center is developing coherent optical pattern recognizers with invariances to various of the six kinematic parameters.[‡] Besides these internal NASA programs (this list is almost certainly incomplete), programs must be developed outside NASA as well if optics is to play a powerful role in these robots. A marriage of work in optical processing both inside and outside NASA will be required for these robots.

One task not being attacked is very rapid (relative to the frame time) nonlinear image preprocessing. We have in mind tasks such as skeletonization, median filtering, feature location, and noise removal. Such tasks are well suited to modern cellular array processors, and the speeds of some of these are essentially fine for the task. On the other hand, if we wish to do many such operations in a frame time (a strong likelihood in view of the iterative nature of many of the algorithms), new technologies may be needed. Also, we would like flexibility to program the cellular array processor to perform noniterative sequences of operations. These tasks may be facilitated by an optical cellular array processor.

2. CELLULAR ARRAY PROCESSORS

Cellular array processors are simply regular arrays of locally interconnected synchronous processors, or cells. There is a well-defined cycle time in which each cell receives information from all its neighbors, performs its characteristic calculation, and has its value replaced by a new value. Normally, there is a one-to-one mapping of cells onto pixels.[1]

We consider here only finite impulse response (FIR) operations, or neighborhood operations.[2] For this preliminary study we specialize to a very small but serviceable 3 × 3 neighborhood in a square array. Furthermore, we specialize to a binary image. Removing both specializations is possible but difficult enough to be a distraction in this initial study. More general and complex optical cellular array processors have been proposed by Tanida and Ichioka.[3,4] By specializing, we can simplify the design considerably.

The approach we use is cell replacement. Rather than explain this method abstractly, we offer some trivial examples that should make the generalization obvious.

Suppose we want to recognize the corners of all objects in the scene. We can do this by replacing the central pixel in a 3 × 3 neighborhood by a 1 if the cell has any of the four patterns shown in Fig. 1. All other patterns will produce a 0 in the center pixel.

[*] D. Cliffone, private communication (1985).

[†] H. K. Liu, private communication (1985).

[‡] R. Juday and Michael Duff, private communications (1985).

Invited Paper SS-105 received Jan. 9, 1986; accepted for publication March 28, 1986; received by Managing Editor March 31, 1986.

$$\begin{array}{ccc} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{array} \qquad \begin{array}{ccc} 0 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{array}$$

$$\begin{array}{ccc} 0 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{array} \qquad \begin{array}{ccc} 0 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \end{array}$$

Fig. 1. 3 × 3 neighborhoods that lead to a 1 in the output for corner recognition.

Now suppose we wish to use a median filter to smooth out noise without blurring "real edges." Then every 3 × 3 pattern with five or more 1's will lead to the center cell being replaced by a 1. All other patterns lead to a 0.

Readers unfamiliar with these concepts might wish to "invent" some "algorithms." For instance, a shift-right-one-cell substitution is easy to discover.

3. GENERAL APPROACH

There are "only" $2^9 = 512$ possible patterns. All we need to do is recognize at most $2^8 = 256$ patterns—the worst case being when 1- and 0-yielding patterns are equal in number.

If we use space-variant pattern recognition, i.e., if we control where the pattern appears, the pattern recognition becomes trivial.

On the other hand, only one pattern at a time can be in any particular location. This suggests that only one output pixel at a time can be generated. As we move the neighborhood from one pixel to the next in raster format, six of the nine pixels stay in the neighborhood, three drop out, and three are added. This suggests a pulsating flow pattern, in fact, a systolic array processor.

Accordingly, we have designed an optical systolic array processor for cellular array processing. Because the processor is pipelined, it generates outputs at a rate proportional to the input rate. A simple systolic cellular array processor of the type to be described can move data out at one-third of the data input rate. If the recognition occurs in parallel (not unreasonable for 256 "channels"), the input is the effective rate limiter.

The rest of this paper shows one way of doing this. Many other ways (some without pulsing, some doing multiple lines at once, etc.) will occur to optics-oriented readers. The method we show was chosen for didactic and constructive simplicity.

Let us label the cells as follows:

$$\begin{array}{cccc} (1,1) & (1,2) & \ldots & (1,N) \\ (2,1) & (2,2) & \ldots & (2,N) \\ \vdots & \vdots & & \vdots \\ (M,1) & (M,2) & \ldots & (M,N)\ . \end{array}$$

Any size or shape of neighborhood can be defined. Somehow padding (pixels around the edges of images) must be defined to allow edge cells to be calculated. A sample neighborhood is the 3 × 3 cell. For instance, the "image" of cell (2,2) [call it (2,2)′] depends on the pattern of 1's and 0's in the subarray:

$$\begin{array}{ccc} (1,1) & (1,2) & (1,3) \\ (2,1) & (2,2) & (2,3) \\ (3,1) & (3,2) & (3,3)\ . \end{array}$$

We now show our proposed method for doing the above neighborhood operations optically. We first map the 2-D neighborhood into a 1-D neighborhood in an appropriate way. For (2,2)′, the neighborhood is

$$\begin{array}{c} (1,3) \\ (2,3) \\ (3,3) \\ (1,2) \\ (2,2) \\ (3,2) \\ (1,1) \\ (2,1) \\ (3,1)\ . \end{array}$$

For (2,3)′, the neighborhood is

$$\begin{array}{c} (1,4) \\ (2,4) \\ (3,4) \\ (1,3) \\ (2,3) \\ (3,3) \\ (1,2) \\ (2,2) \\ (3,2)\ . \end{array}$$

Note that moving from (2,2)′ to (2,3)′ involves moving the existing pixels down three positions and adding three new ones at the top. If a shift occurs in Δt, the pattern is LLLLLLLLRLLRLLRLLRLLR..., where L is the load operation and R is the read operation. Flowing pulsating calculations in essentially identical local processing units have come to be called "systolic."

In the simplest case the data will be flowed in using an acousto-optic cell. For this purpose, we can envision an acousto-optic cell as a shift register of optical transmissions. That is, a set of optical transmission values of 1's and 0's can be inserted into the top of such a cell. They thereafter flow through the cell at a uniform speed.

After the initial pipeline loading, an appropriate nine pixels are present to calculate a new image cell.

Let us use two side-by-side acousto-optic cells (in the same material for convenience) to represent a neighborhood. Thus, the 2-D and 1-D mapping converts

$$\begin{array}{ccc} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 0 \end{array}$$

to

0
0
0
1
1
0
1
1
0.

Our new mapping converts this to

0 1
0 1
0 1
1 0
1 0
0 1
1 0
1 0
0 1.

An optical signal with light of a strength 0 or 1 in this pattern is easy to produce with a two-cell acousto-optic device. Both 1's and 0's in the original cell are now represented as light-on positions.

A recognition spatial mask with transmissions 1 and 0 can be inserted into the optical pattern. Indeed, up to 256 binary matched filters can be addressed in parallel using a typical spatial light modulator (SLM) as a mask.

When an exact match between the input signal and the mask occurs, the optical signal integrated over all nine (18) cells is 9. No other mask can give a signal higher than 8. Thus, even analog optics can yield an extremely low misidentification rate.

Any signal (using multiple masks) greater than, say, 8.5 will give an output 1. All other signals give a 0. Figure 2 shows the sequence of operations just discussed.

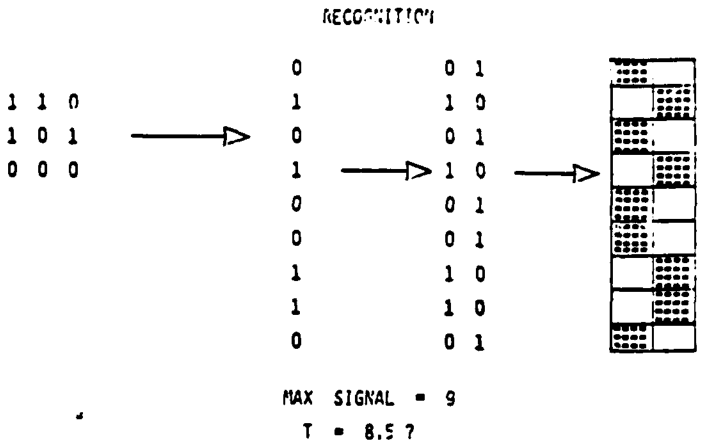


Fig. 2. Sequence of transformation from a 3 × 3 binary array to a unique 2 × 9 array of 9 "on" and 9 "off" cells that, when passed through a matching mask, summed, and thresholded, uniquely identify the particular cell configuration.

Having given an overview, we now backtrack and cover some mechanisms briefly.

Acousto-optic cells are crystals with attached transducers for launching bulk rf acoustic waves into them. The compaction and rarification of the crystal by the sound waves produces an instantaneous diffraction grating that propagates through the crystal at the speed of sound. Any modulation of that rf carrier modulates the diffracted light. Schlieren with optics is used to image only the diffracted portion of the incident light. The effect produced is that of a moving amplitude pattern—exactly what is needed according to the analysis just presented.

Fixed recognition masks can be made photographically, but SLMs allow us to build a real-time reprogrammable optical cellular array processor.

4. ASSESSMENT

This simplest optical cellular array processor inputs data to the acousto-optic bandwidth B and outputs data at a rate B/3. For a frame of 500 × 500 pixels and a rational bandwidth of 90 MHz this leads to 120 frames per second. More complex processors will go even faster.

5. ACKNOWLEDGMENT

This work was supported under contract N00014-85-K-0479, University of Dayton Research Institute.

6. REFERENCES

1. K. Preston, *Modern Cellular Array Processors*, Plenum Press, New York (1984).
2. S. Levialdi, "Neighborhood operators: an outlook," in *Pictorial Data Analysis*, R. M. Haralick, ed., p. 1, NATO ASI Series, Spring-Verlag, New York (1982).
3. J. Tanida and Y. Ichioka, "Optical-logic-array processor using shadowgrams. II. Optical parallel digital image processing," J. Opt. Soc. Am. A 2(8), 1237 (1985).
4. J. Tanida and Y. Ichioka, "Optical-logic-array processor using shadowgrams. III. Parallel neighborhood operations and an architecture of an optical digital-computing system," J. Opt. Soc. Am. A 2(8), 1245 (1985).

H. J. Caulfield is Director of the Center for Applied Optics at The University of Alabama in Huntsville. Widely regarded as a major contributor to optical computing, he has chaired numerous meetings in the field, edited the most widely cited review of the area, and invented many of the basic concepts. He is also well known in holography, where his publications include two books and perhaps the most widely read article on any technical area: the March 1984 cover article of *National Geographic*. Dr. Caulfield is a Fellow of both SPIE and OSA. He served for six years as editor of *Optical Engineering* and is currently associate editor of *Applied Optics*, *Laser Focus*, and *Holosphere*. He has been an officer and honoree of both SPIE and OSA on many occasions.

Optical interconnection based symbolic manipulations

H. John Caulfield, FELLOW SPIE
Center for Applied Optics
The University of Alabama in Huntsville
Huntsville, Alabama 35899

Abstract. Using arrays of optical bistable devices addressed and read out by arrays of holograms, we can make an optical symbolic computer that can learn languages, play games, invent stories, etc.

Subject terms: optical interconnections.

Optical Engineering 25(10), 1179-1180 (October 1986).

CONTENTS

1. INTRODUCTION

While most attempts at optical computing have aimed at numerical processing[1-5] or numerically assisted reasoning,[6-8] little effort seems to have been devoted to symbolic proccssing. In particular, prior work on optical symbolic processors has aimed at drawing inferences from input data, stored data, and rules. This paper is intended to show that optical computers can do far more than that. In principle, they can be taught to speak English, tell stories, play simple games, etc. The particular approach shown here is chosen because, of all the non-inference-drawing schemes I have discovered, it appears to be the simplest to implement optically. No claim to optimality in any sense is made; rather, I hope to draw some attention to this heretofore largely neglected task for optical processors.

By symbols we mean what is meant in normal conversation: letters, words, numbers, events, concepts, etc. We assume that these symbols can be listed, i.e., that they are countable. This means that we could make a list of them and assign a number to each. Our object will be to produce meaningful and useful strings of symbols: sentences, equations, reactions, etc. One of the good features sought is innovation, although we may wish to control its rate of production.

2. JOINT CONTEXT NETWORK (JCN)

The essence of the JCN is to remember, predict, or postulate the next symbol, given the previous N (as well as prior teaching) as the context. We will call N the context depth.

An example may prove helpful. I have just read an article on speech recognition. A typical sentence begins: "The differences between normal and . . . ." With (1) your knowledge of the structure of the English language and (2) the context of that beginning, you have little difficulty in "predicting" that the next word will be "abnormal." Can we make an optical symbolic computer that can do the same? Confronted with a new situation, can we generate an appropriate response based on past learning plus trained "insight"?

A wonderfully readable book by J. H. Andrae, titled *Thinking with the Teachable Machine*,[9] gives the details of the context-driven approach we call JCN. I show here a simple example. Say my task is weather forecasting. I want to know if today will be clear (C), rainy (R), or partly cloudy (P). The past few weeks have been CCCPPRRPCC CCCCRPCCPCCCCCCP. What should I predict next? For N = 2 JCN, we learn

CC → C (the first two symbols implied the third) ,
CC → P ,
CP → P ,
PP → R ,
PR → R ,
RR → P ,
RP → C ,
PC → C ,
CC → C (earlier we had CC → P) ,
CC → C (a popular implication) ,
CC → C ,
CC → C ,
CC → R ,
CR → P ,
RP → C ,
PC → C ,
CC → P ,
CP → C ,
PC → C .
.
.
.
.
.

Our N = 2 context is CP. Based on prior observations, the two

Paper SC-6112 received March 4, 1986; accepted for publication July 1, 1986; received by Managing Editor July 21, 1986.

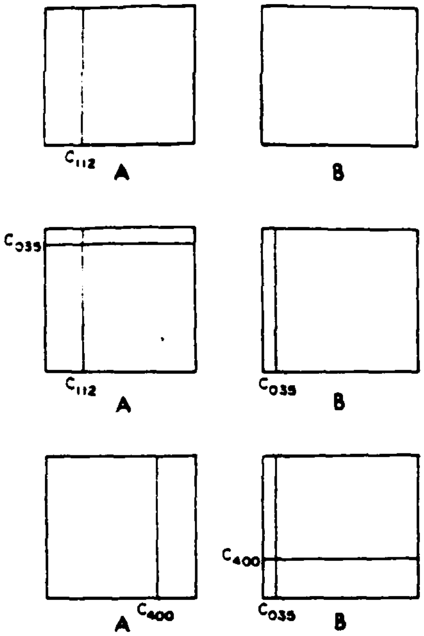


Fig. 1 For context N = 2, we need two bistability arrays (A and B). In (a), we see concept 112 being entered as a column of light for one panel (A). In (b), concept C_{035} is entered on both. Because the C_{112} is still on, the threshold is exceeded at the intersection. This occurs only when C_{112} is followed by C_{035}. In (c), we have added concept C_{400}. Because the C_{035} light is still on, the interaction in B exceeds threshold and indicates that concept C_{035} followed by C_{400} has just occurred. Holograms illuminated by light passing through those intersection points spread light across detectors or other bistable panels if greater than depth 2 is required.

possible predictions for the next observation are P and C. We can choose one at random or seek further context. The N = 3 JCN also would predict P and C, so that is no help. An N = 4 JCN would predict P; that may be our best bet.

3. ENCODING

Since, by hypothesis, the symbols are countable, we can put them in one-to-one correspondence (order unimportant) with the counting numbers 1, 2, 3, ... Suppose, for the moment, that the number of symbols is small, say 512, and the JCN has a context of only N = 2. We use two 512 × 512 arrays of optical bistable devices. Call them A and B.

Figure 1 shows how A and B are addressed if the symbol string is C_{112} C_{035} C_{400}. After the context C_{112} C_{035} is established, a unique intersection occurs in A. The context C_{035} C_{400} is a unique intersection in B. The next intersection occurs in A, etc.

Thus odd-numbered (1st, 3rd, ...) concepts must be horizontal in A and vertical in B, while the reverse is true of even-numbered concepts.

To address a concept, we must know whether it is odd or even. We then either turn on a light beam or deflect a light beam to the proper position. The beam then strikes a hologram, which spreads it as required. Each beam is on for the length of the context (two "read times" in this example). Each beam has strength *I*, and a *threshold of around 1.5 detects* coincidences. The transmitted light or (better) newly generated light from a bistable laser strikes another hologram, which *(a) gives the prediction or memory or* (b) gives the set or weighted set of predictions or (c) sends data on to other processor arrays.

Cascading a number of these systems to use a depth greater than 2 is straightforward.

4. EXAMPLE CASE

Suppose we want a 512 symbol N = 2 JCN. We postulate two 512 × 512 arrays of bistable lasers that emit only if struck with light of power at least I_T. Each symbol can be represented by one of 512 A sources or one of 512 B sources. Each of these 1024 sources is followed by its own hologram, which directs uniform vertical and horizontal light beams toward the A and B bistable laser arrays. The power on each illuminated laser is less than $I_T/2$. Thus, only at the intersection does a laser beam arise. That laser beam, in turn, strikes its own hologram, which causes the light to predict or remember something from the joint conjunction by illuminating one or more of 512 detectors.

Programming or teaching is embodied in the two 512 × 512 arrays of memory holograms. If the output (22 × 23 = 506 plus an extra row of 6) is the joint Fourier transform of both 2-D laser-hologram arrays, the programming holograms are simply properly aligned and spaced gratings.

5. ANALYSIS

This paper is intended to illustrate a new direction for optical computing: symbolic processing. Optical parallelism makes speed independent of the number of symbols processed, although hardware complexity does increase. The JCN speed can be made independent of N (the context depth) by preparing some contexts while reading others. Again, the price is complexity.

6. REFERENCES

1. "Optical Computing," Special Issue of Proc. IEEE (July 1984).
2. J. W. Goodman, A. R. Dias, and L. M. Woody, "Fully parallel, high-speed incoherent optical method for performing discrete Fourier transforms," Opt. Lett. 2, 1-3 (1978).
3. H. J. Caulfield, W. T. Rhodes, M. J. Foster, and S. Horvitz, "Optical implementation of systolic array processing," Opt. Commun. 40, 86-90 (1981).
4. R. A. Athale and W. C. Collins, "Optical matrix-matrix multiplier based on outer product decomposition," Appl. Opt. 21, 2089-2090 (1982).
5. M. Carlotto and D. Casasent, "Microprocessor-based fiber-optic iterative optical processor," Appl. Opt. 21, 147-152 (1982).
6. D. Psaltis and U. Farhat, "Optical information processing based on an associative-memory model of neural nets with thresholding and feedback," Opt. Lett. 10, 98-100 (1985).
7. G. Eichmann and H. J. Caulfield, "Optical learning (inference) machines," Appl. Opt. 24, 2051 (1985).
8. H. J. Caulfield, "Optical inference machines," Opt. Commun. 55, 259 (1985).
9. J. H. Andrae, *Thinking with the Teachable Machine*, Academic Press, New York (1972).

H. John Caulfield: Biography and photograph appeared on p. 827 in the *July 1986 issue.*

Massively parallel optical data base management

H. J. Caulfield
University of Alabama in Huntsville
Center for Applied Optics
Huntsville, Alabama 35899

ABSTRACT

Using Page Oriented Holographic Memories (POHMs) optically addressed Spatial Light Modulators (SLMs), joint transform correlators, 2D or 1D acousto-optic cells, and optically addressable RAMs we can produce a massively parallel optical data base management system.

I. INTRODUCTION

Optical Data Base Management Systems (DBMS) operating with massive parallel read in from memory, query, and read-out to electronics would offer huge advantages over the current bit-oriented or hoped-for byte-oriented systems if

- the data base is too massive for conventional DBMS systems,
- the access time required is too short for conventional DBMs systems, or
- (preferably for optics) both.

When there is need to search huge data bases very fast, we are automatically in the big system domain which will not exclude fairly complex optical systems. Therefore, the objective of this work is to explore massively parallel optical DBMS.

II. KEY INGREDIENTS

The key ingredients are optical systems for

- massive parallel read into an optical system from a large data store,
- parallel query on the whole "page" in the optical system,
- parallel read out from the optical system to output electronics, and
- parallel "intelligent" control of the operation. All of these tasks can and probably should be performed optically.

For brevity, we deal here only with the first two tasks. A separate paper will discuss the latter two.

III. PARALLEL READ IN

Obtaining whole pages of data in parallel is the domain of page oriented holographic memories or POHM (1). Each page is represented by its own spatially discrete hologram on a large substrate. Storing a 256 x 256 page requires a roughly one millimeter hologram. We can store 10^6 of these on a 1m x 1m substrate. The holograms are accessed individually by deflecting a laser beam to the selected hologram. Whichever hologram is selected produces its page at the same physical location. There it strikes an optically addressed Spatial Light Modulator (SLM) which reads the full page into the optical system. This is shown in Figure 1.

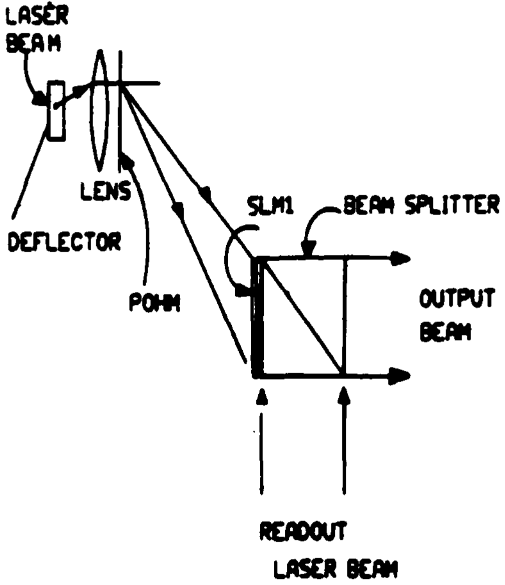


Fig. 1. The deflector addresses a single hologram from the POHM which writes a page of data in parallel onto the output laser beam.

IV. PARALLEL QUERY

The first step in parallel query is to restrict the field of regard to items of interest. We may have the data base arranged in columns such as

Ordinal Number	Family Name	First Name	Second Name	SSN

Our task is to find all people with the second name "John" with 54 as the fourth and fifth numbers in their SSN (Social Security Number). Using an electronically addressed SLM we illuminate only the columns for second names and fourth and fifth numbers as shown in Figure 2.

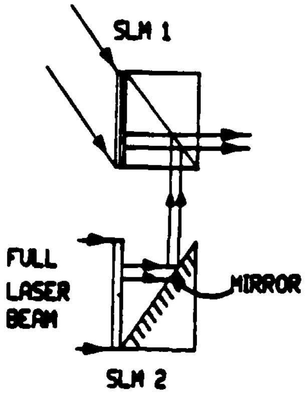


FIG. 2. A SECOND SLM (SLM2, SHOWN HERE AS TRANSMISSIVE) IS USED FOR SELECTIVE ILLUMINATION OF THE OUTPUT FROM SLM1.

The next step is to search for matches with

———	————	JOHN	--- 54 ----

We do this by template matching in parallel using a joint transform correlator as shown in Figure 3.

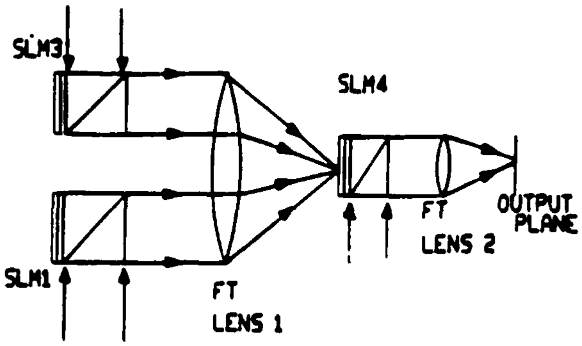


FIG. 3. WE CAN USE AN ELECTRONICALLY ADDRESSED SLM (SLM3) TO WRITE A REFERENCE PATTERN TO BE MATCHED WITH THE LIGHT COMING FROM THE FORM-DRIVEN SLM (SLM1). FOURIER TRANSFORM LENS 1 JOINTLY TRANSFORMS BOTH OUTPUTS ONTO 4 SLM WHERE THEY ADD COHERENTLY. ANOTHER READOUT BEAM ADDRESSES SLM4 WHICH PRODUCES BRIGHT LIGHT ON THE OUTPUT PLANE WHERE THE REFERENCE PATTERN IS MATCHED WITH A PORTION OF THE INPUT BEAM (FROM SLM1).

V. READOUT AND CONTROL

We will only hint at these items here. The selected rows output must be accumulated continuously on a parallel read in RAM for subsequent use. Figure 4 shows the basic concept.

Control is the most difficult part if we wish to replace exhaustive search with heuristic search. In a later paper we will show an adaptive optical neural network suitable for this purpose.

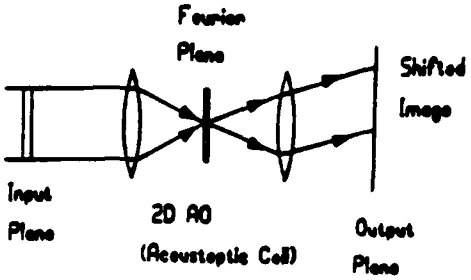
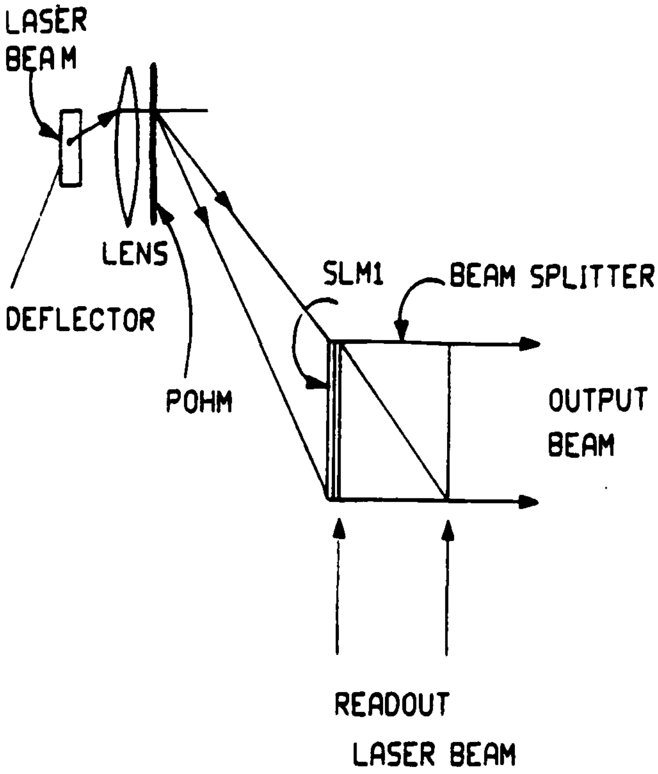


Fig. 4. The 2D AO cell shifts the image in any direction by changing the direction of light in the Fourier plane.

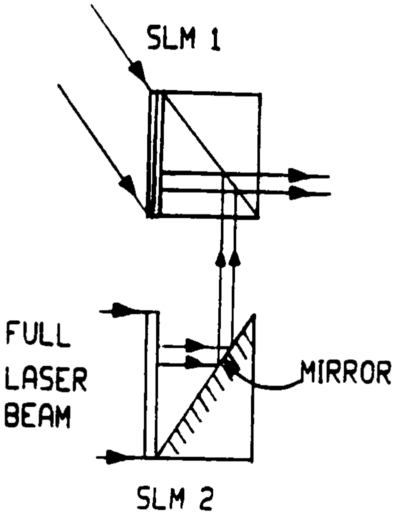
REFERENCES

1. P. M. Smits and L. E. Gallaher, Bell Systems Technical Journal, 46, 1267-1278 (1967).



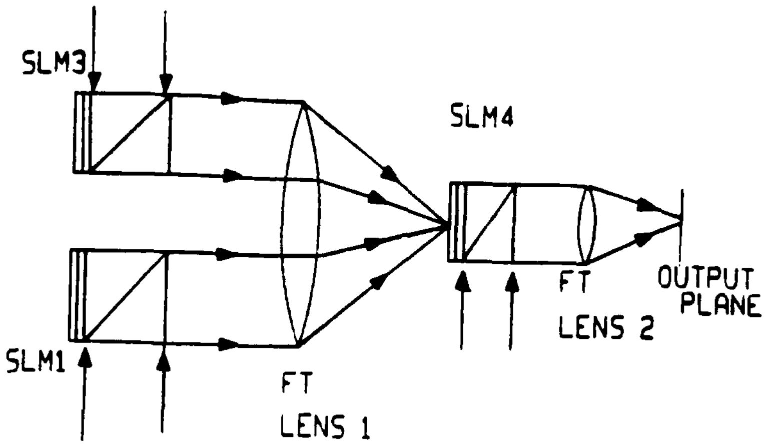
The deflector addresses a single hologram from the POHM which writes a page of data in parallel onto the output laser beam.

Figure 6 PAGE SELECTION AND READOUT



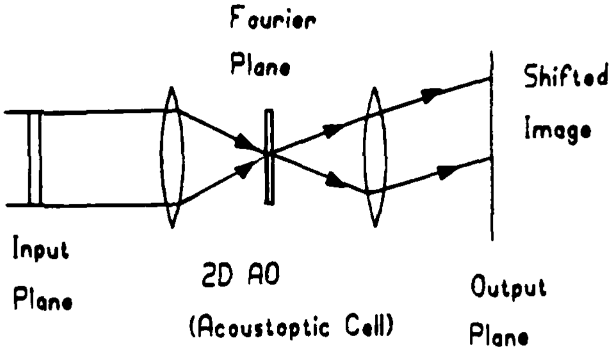
A SECOND SLM (SLM2, SHOWN HERE AS TRANSMISSIVE) IS USED FOR SELECTIVE ILLUMINATION OF THE OUTPUT FROM SLM1.

Figure 7 OUTPUT SELECTION



WE CAN USE AN ELECTRONICALLY ADDRESSED SLM (SLM3) TO WRITE A REFERENCE PATTERN TO BE MATCHED WITH THE LIGHT COMING FROM THE POHM-DRIVEN SLM (SLM1). FOURIER TRANSFORM LENS 1 JOINTLY TRANSFORMS BOTH OUTPUTS ONTO 4 SLM4 WHERE THEY ADD COHERENTLY. ANOTHER READOUT BEAM ADDRESSES SLM4 WHICH PRODUCES BRIGHT LIGHT ON THE OUTPUT PLANE WHERE THE REFERENCE PATTERN IS MATCHED WITH A PORTION OF THE INPUT BEAM (FROM SLM1).

Figure 8 PATTERN SELECTION



The 2D AO cell shifts the image in any direction by changing the direction of light in the Fourier plane.

Figure 9 IMAGE SHIFTING

Optical database/knowledgebase machines

P. Bruce Berra, Karl-Heinz Brenner, W. Thomas Cathey, H. John Caulfield, Sing H. Lee, and Harold Szu

In this paper we discuss various aspects of databases and knowledgebases and indicate how optics can play an important role in the solution of many of the previously unsolved problems in this field.

I. Introduction

A database is a collection of interrelated data and during the past 10 years the word database has become somewhat of a household word. This has occurred because of the ever increasing use of databases and the realization of their considerable influence on our daily lives. They are indispensable to airlines, automobile companies, grocery chains, department stores, hospitals, colleges and universities, local and state governments, and the federal government. Their existence is so important that in many organizations the database is considered a resource as are personnel and raw materials.

Since software database management systems (DBMS) often exhibit poor performance, considerable research has been devoted to specialized hardware devices, called database machines, to seek performance improvement. These devices take advantage of the significant advances in electronic hardware technology by moving software functions into hardware and possess considerable parallelism.

One of the most rapidly growing fields of artificial intelligence (AI) is knowledgebase systems. A knowledgebase consists of rules and facts about particular domains of interest, and knowledgebase management systems are concerned with inferencing on the knowledgebase, as well as other functions. The most well-known system of this variety is the expert system. Current expert systems exist or are being constructed in business, medicine, national defense, and engineering.[1] There has been relatively little research directed to the development of knowledgebase machines.[2]

There is a great deal of commonality between database systems and knowledgebase systems. In fact, there is considerable research and development currently going on which is aimed at the integration of the two types of system. One of the results of this integration is the requirement for increased performance of the integrated system over either of the individual systems. Database machines have had as their objective an increase in the performance of the database system primarily in addressing problems that have a very large database and/or a real time requirement. While the performance of these systems has been improved somewhat, they have not yielded the results that were desired.

In dealing with the types of very large and/or real time problem that we are interested in, it is natural to look to optics for possible solutions. This is due to the large storage capacities available through the use of optical media and the inherent speed and parallelism of light. Thus we examine here the potential performance improvements obtainable from optical database/knowledgebase machines.

We begin by considering database management, database machines, and knowledgebase management. We then present a paradigm for analyzing the potential advantages of optics. This is followed by sections on storage strategies, access strategies, and processing of data prior to conversion to electronic form. Finally, we summarize our analyses and cite some future directions that hold considerable promise.

II. Background

A. Database Management

A database management system is a software program that is concerned with the task of controlling and

P. B. Berra is with Syracuse University, Department of Electrical & Computer Engineering, Syracuse, New York 13244-1240; K.-H. Brenner is with University of Erlangen-Nuremberg, Physics Institute, D-8520 Erlangen, Federal Republic of Germany; W. T. Cathey is with University of Colorado, Center for Optoelectronic Computing Systems, Boulder, Colorado 80309-0425; H. J. Caulfield is with University of Alabama, Center for Applied Optics, Huntsville, Alabama 35899; S. H. Lee is with University of California, San Diego, Electrical & Computer Engineering Department, La Jolla, California 92093; and H. Szu is with U.S. Naval Research Laboratory, Washington, DC 20375-5000.

Received 13 January 1989.
0003-6935/90/020195-11$02.00/0.

managing the database as a resource independent of the computer hardware that hosts it and application programs that interface with it. The DBMS must have the facility to establish the database within the system in response to the database designers. The DBMS must make the data available to a wide variety of users ranging from external application programs to a casual user posing a particular query. Inevitably, the database must be updated. That is, new data must be added, old data must be deleted, and existing data must be changed. Thus, the DBMS must also have the capability for performing these updates. In fact, many databases (e.g., airline databases) have almost as much update activity as query activity. Of course there are many types of database that have limited or controlled update activity (e.g., various forms of text databases).

The DBMS must provide the facility for insuring the integrity of the database. This is obtained through various consistency checks and backup and recovery systems. Finally, the DBMS must regulate access to the database to protect it, the system itself, and the privacy of users.

It is not surprising that DBMSs which furnish all of this functionality tend to be expensive and require considerable computing resources to be effective. While it is true that one can purchase DBMS for personal computers, these systems do not possess all the functionality discussed above and are therefore not the focus of this paper. Rather, we are concerned with systems that must deal with very large databases (VLDB) (hundreds of gigabytes) and/or have a real time requirement (1 s or less response time).

Since the DBMS is just another application program, albeit with considerable subprograms, it must adhere to normal execution procedures just as other programs. The database user (a human user or application program) interacts with the DBMS through a query language (or other language) to accomplish a task. The DBMS must interact with the operating system to obtain data from the database which is stored on the computing system's secondary memory. Since the operating system must satisfy a large number of types of user, the size of the block of data retrieved from disk is optimized for all users and is thus fixed. The block of data is placed in main memory and turned over to the DBMS which sifts through it to find what it wants. There may be little data of interest to the DBMS due to the organization of the data and type of query. Thus, the DBMS may have to ask the operating system for many pages of data to satisfy a query. This repeated access to secondary storage considerably degrades the performance of the DBMS since the access time to the disk is about one million times slower than access to main memory. This disparity is called the access time gap.

B. Database Machines

A typical structure for a database machine is that of a frontend–backend system. That is, the user interacts with a sequential computer host which transforms the request into a series of commands that can be executed by the backend database machine. The database machine handles all database functions and returns results to the host which then passes them on to the user. There are many advantages of this configuration including removal of dependence on the operating system, reduction in the number of functions performed (i.e., the database machine only executes database functions), optimization of execution of certain functions (i.e., special hardware for relational operations), mitigation of the access time gap problem through parallel access to multiple disks and, in general, the advantages one has in solving a more narrowly defined problem.

There are also disadvantages to this configuration. These systems tend to be more costly and less available than sequential DBMSs. There are dozens of university and industry database machine projects but there are just a few commercially available products.[3,4] However, there are hundreds of sequential DBMSs. If the problem being addressed is basically sequential, no amount of parallelism will help; in fact it may even degrade performance beyond that of a sequential DBMS. For example, if a query consists of several subqueries each of which depends on the result of the previous subquery, the traffic across the host–database machine interface will significantly degrade the performance of the system.

As was previously pointed out, we are concerned with VLDB and/or real time requirements; database machines also address these requirements. Thus, while there may be hundreds of sequential DBMSs available, only the few of them residing on major mainframe computers are able to address the requirements. The comparison then comes down to large mainframe systems with DBMSs vs database machines.

A problem addressed by most database machine designs is that of parallel access to magnetic disk. Their approaches are only partially successful since the difficulty basically lies with the mechanics of the disk. The speed of rotation and the extremely small distance between the read–write head and the disk surface are such that the sustained transfer rate of large commercially available magnetic disks tops out at ~3 Mbytes/s. The exception is multiactuator, multihead disks which can increase this rate but at a considerably greater cost. Because of this current limitation, database machine architects have designed their systems to be able to accept and process data from magnetic disks at these rates. However, if data rates were available at 300 Mbytes/s, these database machines would have considerable difficulty dealing with the situation. They would become compute bound rather than I/O bound as is currently the case. We will return to this point later in the paper.

C. Applications Requirements

In examining data processing in general and database processing specifically for their applicability to both near and far term applications, one is faced with a varied and dynamic set of both operational problems

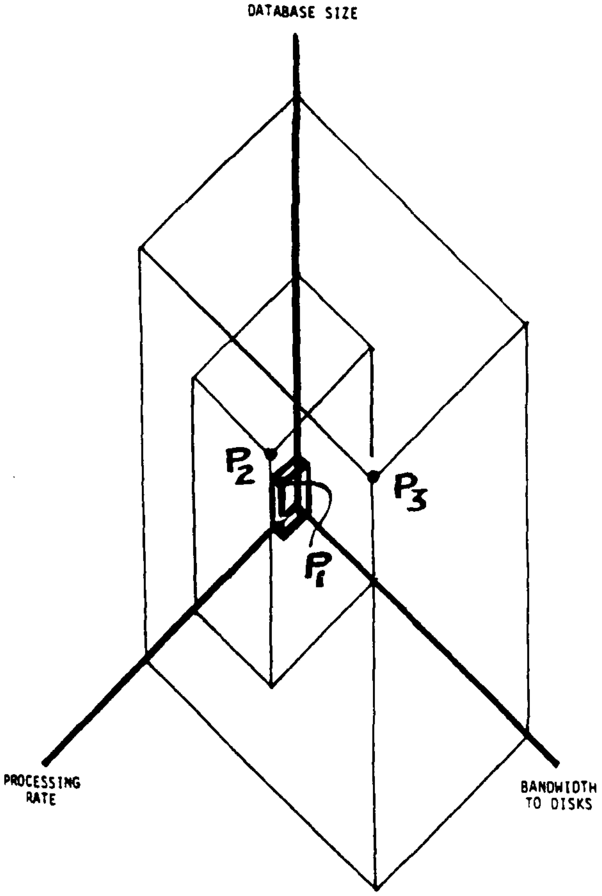


Fig. 1. Size, bandwidth, and processing rate.

as well as technological solutions. Rather than performing the analysis on each of these problem/solution/timeframe tuples, it is desirable to develop a generic analysis technique. Specifically, the technique transforms the system requirements onto three independent axes: database size, bandwidth of communications to the disks, and processing rate as shown in Fig. 1. The size of the database is shown on the vertical axis and is measured in bytes. The bandwidth is measured in megabytes per second and provides a measure of what is required to access disk storage in solving a variety of problems. Another measure of bandwidth is the amount of query input data to the system. However, this input communication bandwidth results in considerable accesses to the disks as well as increased processing requirements. Thus we will use bandwidth to refer to internal I/O bandwidth. The third axis represents processing rate. In multiuser environments a large query load generally results in a large processing load. However, relatively short complex queries can also result in large processing loads. But, in general, database management places more stress on I/O rather than on processing.

As a first example, suppose a sensor has measured the signature of a radar and one must match this signature against a library of known signatures in order to take an appropriate response. Suppose that there are 10,000 known radar emitter signatures, and a new emitter appears every 100 s. The required processing time is not related to the input rate. However, this application may require a match in 10 ms to make a timely response. While neither the measured signature, nor the signature database has 100% validity, we would like the process to be error free. Typically, the measured signature will have certain parameters that are correct and some that are in error. In a specific report, the degree of error is uncertain and the matching progress must allow partial/probalistic matches. Thus, a single input may result in many internal processes. This application is depicted as P_1 in Fig. 1.

As a second example, a contact comparison application might have a database of the order of 10^9 bytes, a contact report rate of 1/s, and a match requirement of 2/s. This application is depicted as P_2 in Fig. 1.

As a third example, a monopulse signal sorter takes a set of measured parameters on a single radar pulse and attempts to identify the source of the pulse from a database of emitters known to be currently hearable. In a complex environment, there may be 1000 emitters hearable at one site, each emitting 1000 pulses/s. The expected report rate is then 10^6 pulses/s. The current database consists of the 1000 emitters. The time to locate a match must be $<1\ \mu s$ just to keep up with the input rate. The front end sensor may produce erroneous results, for example, when pulses from two different emitters are overlapping in time and may miss pulses that are too weak to be detected. This will create two problems. First, there will be holes in the database which will not be filled in without additional manipulation of the database; and second, there will be residual reports that do not correspond to any real emitter. This application is depicted as P_3 in Fig. 1.

If one had an infinitely fast serial machine with, say, 100 Gbytes of memory, one could solve all the example problems, but not necessarily in a cost-effective manner. Similarly, a very large content addressable memory could be used to solve all the problems but would be a gross overkill of certainly the first example. The challenge, then, is to develop techniques to handle the above range of examples, which are generic functions of the application and the state of technology, to invest resources in the minimum number of architectures to solve the collective database processing tasks.

An important consideration is the cost of various technological alternatives. For example, one can purchase off-the-shelf chips at about \$100/Mbyte and disk memory at about \$50/Mbyte based on commercially available 256k RAM chips and 100 Mbyte disks. The advantages and disadvantages of choosing chip or disk with regard to database size, I/O communication, and processing rate can be argued in many ways depending on the task at hand. RAM memory has a very fast access time (200 ns) but is volatile while disk memory is low (30 ms access time) and nonvolatile. In contemporary systems, RAM appears in limited quantity while disks appear in large quantity. But, with the relatively recent reduction in the cost of RAM, this proportion is expected to shift dramatically in favor of RAM.

Returning to Fig. 1, no single architecture is best for all the tasks at hand. Specifically for database problems, it depends on at least the three characteristics given on the axes. For example, with P_1 a moderate amount of data with little need for bandwidth or processing describes the problem. In this case, there is little requirement for special architectures for processing the data or increasing the bandwidth to disk.

With P_2, there is a need for the storage of large amounts of data, so considerable disk storage is required. However, there are also moderate requirements for bandwidth to disk as well as processing power. In this situation, parallel computer architectures may be employed with some usefulness. We may employ a variety of such architectures from single instruction–multiple data stream to multiple instruction–multiple data stream. The point here is that we must now look to parallel architectures to keep up with the processing load. Also, we must be able to access considerable amounts of data so the bandwidth to secondary storage must be high. This will require parallel access to disks perhaps along the lines of commercially available database machines.

The third problem, P_3, is much more difficult to deal with since it has such stringent requirements on all dimensions. Optical storage may help with this problem due to its high density, but data will have to be retrieved from the disk at a faster rate. Optical interconnects will definitely help because of high bandwidth, and optical or electrooptical processing may offer some solutions in the future.

D. Knowledgebase Management

Knowledgebase systems are composed of a knowledgebase of rules and facts and an inferencing mechanism that is used to respond to queries using the existing knowledgebase. In the case of expert systems, the objective is to capture the knowledge of experts in particular domains and make it generally available to nonexperts. Various knowledge structuring techniques include semantic networks, production rules, logic, and frames with the LISP and Prolog languages in common use. Current expert systems tend to focus on narrow domains, have small knowledgebases and, therefore, have limited application. As these systems expand and more general applications are considered, increasing demands will be placed on the management of the knowledgebase. The database of rules (called the intensional database) will become large but the major management problem will be in the access, update, and control of the database of facts (called the extensional database).

The above considerations have led to many research efforts aimed at the interface and eventual integration of knowledgebase systems and database systems. Some systems are currently available that provide an interface between a knowledgebase system and the DBMS. While this allows for the management of larger knowledgebases, the performance of such systems is less than desirable because of the slow interface and the duplication of functionality of the two systems. Another approach that is being taken is to extend the capabilities of the knowledgebase system through the addition of secondary storage management. Still another approach is the addition of inferencing functionality to existing DBMS. All these approaches are headed in the direction of an integrated knowledgebase management system (KBMS) that possesses the capabilities of both systems. However, when viewed from a performance perspective, KBMSs will place even more demands on the underlying data management structures. That is why it is imperative that we look to other technologies such as optics for possible solutions.

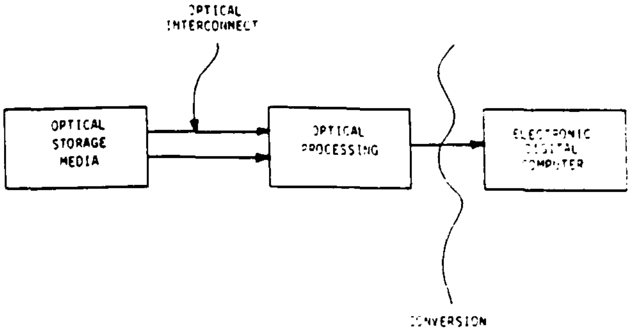


Fig. 2. Optical/electronic paradigm.

III. Hierarchical Structure of Processing

The state of the art of electronic computing enjoys considerable maturity. In contrast, optics as applied to digital computing is very young and has yet to make its mark. In assessing how optics may help database and knowledgebase management, it seems clear that the most impact will be felt at the lowest level. Thus, the approach that we have taken in this paper is an optoelectronic one in which we start at the very lowest level and progressively move toward conversion to electronics as indicated in Fig. 2. We examine various types of optical storage media and devices to assess their potential for use in database and knowledgebase management. As will be discussed later, the potential exists for enormous data rates from optical storage. Since electronic database machines are designed to deal with magnetic disk transfer rates, they will not be able to handle these increased rates. This dictates that we keep the data in optical form and do as much processing as we can prior to converting to electronics. We will discuss the type of processing that can be done later in the paper. However, our objective is to process the optical data to the fullest extent possible so that, on conversion, the data rate will be within the capabilities of the electronic computer but more content rich. In this way we hope to increase the performance of the system without disturbing the large investment in system and user software.

IV. Storage and Processing Strategy

A. Optical Disks

By far the most popular form of optical storage is optical disks. These range from CD-ROMs to very large disk units which allow for massive storage of data.[5,6] Optical disks have a far greater capacity than their magnetic counterpart but have a much slower access and transfer rate. This is primarily due to the mass of the read head and the slower revolution rate. However, while magnetic disks appear to be approaching technological limits with regard to access time and transfer rates, optical disks have great potential for vast improvements. This is true primarily because of the relatively large distances between the read mechanism and the disk surface. Through multibeam reading the potential exists for massive data transfer rates of the order of 300–500 Mbytes/s, a full 2 orders of magnitude over current magnetic disks.

B. Content Addressable Memory

There have been several laboratory demonstrations of associative memory.[7–9] Variations of this approach may be applicable for a content addressable memory.

One technique, using holographic memory, is to store the data holographically and to provide feedback with gain. This system, illustrated schematically in Fig. 3, operates as follows: The combination hologram and resonant structure has transverse resonant modes that are defined and limited by the images stored in the hologram; that is, only resonant modes corresponding to a holographic image are possible. A partial image is fed into the hologram, which causes a more complete image to be reconstructed from the hologram. This image receives gain in the nonlinear medium, and the resonant structure resonates with that image. If a portion of another image is input into the system, the transverse mode associated with that image becomes dominant, and after a few passes around the closed path with gain, that image is fully recalled. If more than one stored image contains the image portion that is fed into the system, one image will start to dominate due to greater correlation with the input or the characteristics of the noise in the system and, once the system locks onto that mode (image), it stays on that image.

In a content addressable memory, it is desired that all sets of data with a common part be retrieved. For example, if the word Colorado is input to a database of Optical Society members, it should be possible to retrieve all appropriate names of members in Colorado either in series or parallel. In a database/knowledgebase system, it is necessary that a partial input into a file retrieve all components of that file having that partial input. In currently demonstrated associative memories, only one component would be retrieved and the one retrieved would differ from time to time depending on the noise state of the system. To change the associative memory into a content addressable memory suitable for database/knowledgebase systems would require some means of recalling them all. A perturbation of the system would be necessary to move the resonant system to another transverse mode and another output in the common file. Thus far, no such demonstration has been made.

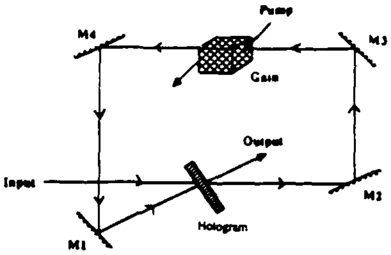


Fig. 3. Content addressable memory with feedback and holographic storage.[7]

Another unresolved problem with holographic associative memory (or any holographic storage system, for that matter) is that the possible number of stored images predicted by current theory is several orders of magnitude greater than has been achieved experimentally. More accurate analyses and simulations are needed before these discrepancies can be resolved.

C. Page Oriented Holograms

1. *Storage*

Most massive database and knowledgebase systems store data on magnetic or optical disks and employ indexing techniques to avoid or minimize disk accesses. Various clustering and accessing techniques are used to reduce response time. Even so, when the joint requirements of very large databases and very short response times are imposed, existing technologies degrade considerably. In these cases, the ability to call forth and operate on large pages of data in parallel would offer a profound advantage over serial operation. Some of the issues discussed below are also addressed in Ref. 10.

The basic concept of page-oriented holographic memory (POHM) is quite simple. Many small spatially discrete holograms are recorded on a single substrate. Some are constructed such that whenever a laser beam is deflected to one of these holograms, the output 2-D image falls on a common surface for all holograms. Of course, the whole 2-D image arrives essentially in parallel. A 1-mm hologram, properly made, can store an array of 10^4–10^6 bits which is a page. An electrooptic or acoustooptic deflector can address any of these stored pages very rapidly (10^{-4}–10^{-6} s). Access tme is limited by laser deflection times (10^{-9}–10^{-6} s) or paallel readout mechanism response time

[10^{-4}–10^{-2} s for currently available spatial light modulators (SLMs)]. Using the worst (best) case numbers, we can recall 10^4 (10^6) bit pages in any order from among 10^4 (10^6) such pages at a rate of 100 (10^4) pages/s. Thus the capacity of POHM ranges from a few megabytes to over a terabyte while the transfer rate ranges from <1 Mbyte/s or >100 Gbytes/s. We can place an optically addressed SLM at the output as an image amplifier to read the page into the optical system in parallel. Figure 4 shows this basic system along with the laser readout system for the SLM.

What the SLM produces is a modulation pattern (in intensity, phase, polarization, etc.) but light comes only from the portion of the SLM which is illuminated. Thus if we illuminate the SLM addressed by the POHM with light from a second SLM (electrically addressed, intensity modulated), we can restrict entry into the optical system to those portions of the page of immediate interest. Figure 5 shows this part of the system. For read-only POHMs, photographic or other conventional storage methods can be used.

Multiplexed holograms can also be stored in 3-D photorefractive crystals.[11,12] Two schematics of photorefractive memory are shown in Fig. 6. In Fig. 6(a), the *i*th image is stored by interfering the input image with the reference (pump) beam when the photorefractive crystal is rotated to a specific angular position. To read out the *i*th image from the photorefractive memory, the input is turned off and the reference beam turned on when the crystal is rotated to the specific angular position. In Fig. 6(b), the *i*th image is stored by interfering the input with a reference beam of the *i*th phase code. (The photorefractive crystal need not be rotated in this alternate scheme.) To read out the *i*th image from the photorefractive memory, the input is turned off and the reference beam of the *i*th phase code is turned on.

Presently, photorefractive crystals require milliseconds to store a hologram. The hologram writing time can be reduced by using higher intensity beams. It can also be reduced for strontium barium niobate (SBN) by applying an electric field across the crystal. Research for reducing the hologram writing time is under way by increasing impurity doping levels in the photorefractive crystals. To retrieve a stored image from a photorefractive memory, the time required can be much shorter than milliseconds and is determined by how fast the photorefractive crystal can be rotated to the desired angular position in scheme (a) or how fast the reference beam can be switched from one phase code to another in scheme (b). Using SBN:60, hundreds of page-oriented holograms can be stored and retrieved in real time.

Holographic storage is far from perfected despite many millions of dollars of effort expended all over the world in the 1970s. Uniformity among output pixels is seldom better than 10–15%, signal-to-noise ratios can be low, but, outside the Soviet Union, little work has been performed on POHMs this decade. Great improvements arising from subsequent advances in holography and SLMs may be expected.

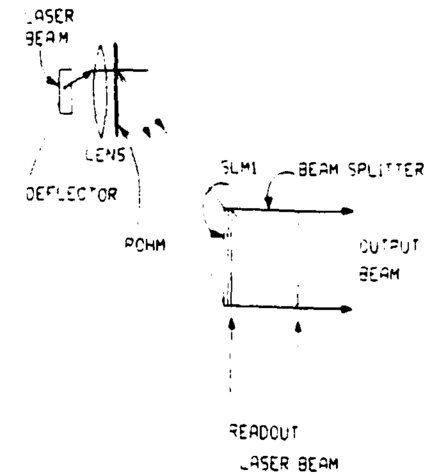


Fig. 4. Page selection and readout. The deflector addresses a single hologram on the POHM and a page of data is written in parallel onto the output laser beam.

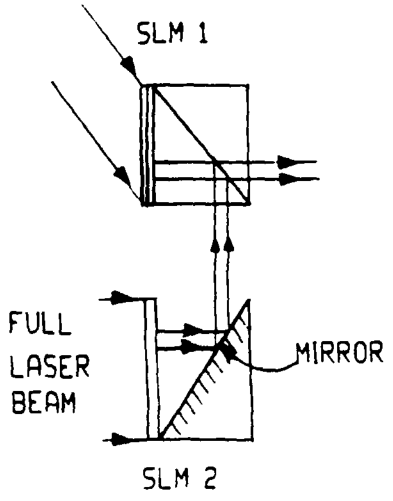


Fig. 5. Output selection. A second SLM (SLM2, shown here as transmissive) is used for selective illumination of the output fro SLM1.

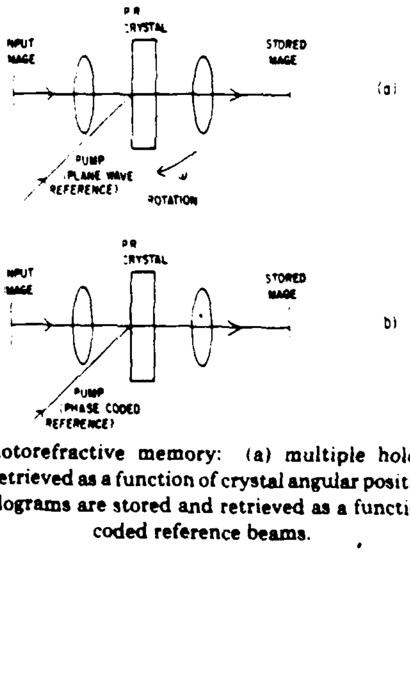


Fig. 6. Photorefractive memory: (a) multiple holograms stored and retrieved as a function of crystal angular positions and multiple holograms are stored and retrieved as a function of ph. coded reference beams.

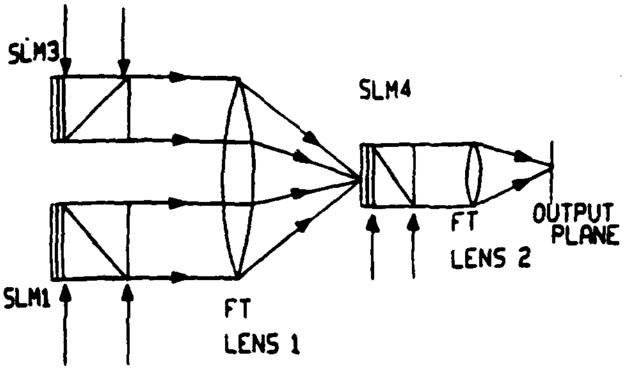


Fig. 7. Pattern selection. We can use electronically addressed SLM3 to write a reference pattern to be matched with the light coming from POHM-driven SLM1. Fourier transform lens 1 jointly transforms both outputs onto SLM4 where they add coherently. Another readout beam addresses SLM4 which produces bright light on the output plane where the reference pattern is matched with a portion of the input beam from SLM1.

2. *Processing*

It will be of interest to search the illuminated portion of the retrieved page in parallel for space-invariant pattern recognition. Fourier optics is known to be excellent for this purpose if we know ahead of time what pattern(s) we want to recognize and prepare appropriate pattern recognition masks. While this technique will be useful in certain cases, it offers insufficient flexibility for general DB/KB purposes, so we must use other techniques such as joint transform correlation.[13-15] To do joint transform correlation, we generate a reference pattern on yet another SLM and use one lens to jointly Fourier transform both images which must be illuminated by the same laser in such a manner that they are mutually coherent in the Fourier plane. There they strike yet another SLM which is read out by yet another laser beam. That laser beam, after reflection from the SLM, is again Fourier transformed to produce an ouptut which resembles the input page but is bright only where the reference pattern appears in the page. This output pattern must be thresholded optically (in parallel) or electronically. Figure 7 shows this arrangement.

Finally, data must be copied from the page onto a scratchpad memory. Where possible, this too should occur in parallel. If we assume that we can accumulate data over time in parallel on a 2-D charge-coupled device (CCD) array for eventual CCD readout the problem becomes one of illuminating only the right part of the page (a problem discussed earlier) and deflecting that light to the right part of the detector array. A fast (microsecond) 2-D acoustooptic image scanner should be ideal for this task. Figure 8 shows how this variable spacing grating can be used in this manner.

Omitted from this discussion of basic methods are drawings of how all the parts fit together in one system.

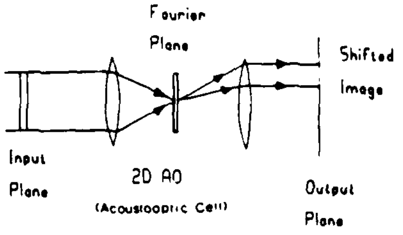


Fig. 8. Image shifting. The 2-D AO cell shifts the image in any direction by changing the direction of light in the Fourier plane.

We believe that this will be complicated but perfectly feasible using beam splitters (amplitude and polarization), reflex mirrors, multiple POHMs, etc. The system, at least initially, will be large and expensive, but the users of very large DB/KB systems are used to size and cost now. What optics adds is high speed.

3. *Two-Dimensional Access*

It is possible, in principle, to move both the medium and the beam in such a way as to use the fastest available 1-D scanners, e.g., chirped acoustooptic cells, with much slower medium translators to interrogate a 2-D data array at a bit rate approaching that of the fast scanner. Suppose an N-bit fast-scan horizontal pattern is available. Let the medium move vertically at a speed S_m. Let the fast acoustooptic scan speed be $S_a \gg S_m$. Then, by tilting the scan direction at an angle

$$\theta_S = \tan^{-1}(S_m/S_a) \qquad S_m/S_a \ll 1$$

from the horizontal we can sweep out a horizontal path at speed S_a. By making rows correspond to attributes or to objects in a relational database, this method could allow up to gigahertz access to the interesting part of a database.

By far the fastest access to a random bit and the most bits read out in parallel result from page-oriented holographic memories discussed earlier[16] and shown in Fig. 4. Let there be H^2 holograms ($H \times H$ array) each presenting a $B \times B$ bit array to the SLM when illuminated. The maximum deflection time is t_p. Clearly, any particular bit from the H^2B^2 number of stored bits can enter the optical system in an access time

$$T = \max(t_p,t_s),$$

where t_s is the SLM response time. Probably, $T = t_s$ can now be 10^{-6} s. For $H = B = 10^3$ (a very large POHM since the individual holograms must be 1-2 mm), we have 10^{12} bits accessible in 10^{-6} s or 10^{18} bits/s. Even if we immediately convert the data to serial format, we still have access at 10^{15} bits/s. Nevertheless, the best course, if feasible, is to keep the page operations parallel and, hence, optical for as long as possible.

Another approach is to select one of an array of holograms as before but allow each hologram to store multiple images.[17] If the images are angularly multiplexed, a 2-D acoustooptic cell at or imaged onto the hologram can allow selection of the desired wavefront. If the images are wavelength multiplexed we must adjust the deflector and tunable source jointly. A hologram storing N images reduces the POHM area by N. However, multiplexed holograms experience reduced signal-to-noise ratio or dynamic range.

D. Spatial Light Modulators

There are presently several spatial light modulators (SLMs) which exhibit optical memory characteristics and may be considered for page-oriented memory applications. They are microchannel spatial light modulators,[18,19] ferroelectric liquid crystals,[20-23] multiple quantum wells,[24-27] silicon-electrooptic modulators,[28] and thermoplastics.[29]

1. *SLM Storage*

Fast, high density reprogrammable electronic memories are widely available. The electronic memory can be divided into $N(n \times n)$ cells in an array format. If optical input and outputs in the form of phototransistors and optical modulators, respectively, can be added to each of the N cells of the memory array, we can obtain an optically accessible N-port memory SLM where N is the number of optical input–output ports. This memory will be page oriented because all N-ports arranged in the array format can be accessed in parallel. The number of memory circuits in each cell will be the depth of the N-port memory.

To provide the optical inputs and outputs for the N-port memory, research on phototransistor design and silicon-electrooptic material integration has been performed.[28,30-32] With PLZT as the modulator material, it is estimated that memory access time of 1 μs is attainable. Low loss polarization switching at microsecond rates has been demonstrated with ferroelectric liquid (FLC) crystals[20] and photo-addressed FLC SLMs have been demonstrated.[23] New electrooptic materials such as organic polymers and GaAlAs or InP multiple quantum well structures[26,27] are currently being studied for access time improvements.

Depending on how the electronic memory in each cell is organized, the N-port memory can be accessed by address or by content. Depending on which and how many cells of the N-port memory are activated, pages of information can be retrieved in parts or in their entirety.

2. *Processing*

It is also possible to perform logic using the SLM.[33-36] If many (or some) of the memory circuits in each cell of the N-port memory SLMs are replaced by logic circuits, we obtain an N-port processing SLM, which can combine the processing power of silicon electronics and the communication or interconnection capability of optics. Depending on the design of logic circuits in each cell, important processing operations such as comparison and matching between new and stored data can be performed in parallel. Furthermore, if the N-port processing SLM can serve as the input of an optical matrix–tensor multiplier, we can perform parallel search for artificial intelligence.[37,38] By allowing parallel readout from mass optical storage to address in parallel the $n \times n$ ports of an N-port processing SLM (with or without an optical matrix–tensor multiplier attached), many well-defined processing functions can be performed at high speed. Then, the electronic data converted from such processed optical data will have much lower data rates on the average, and the data will be much richer in information.

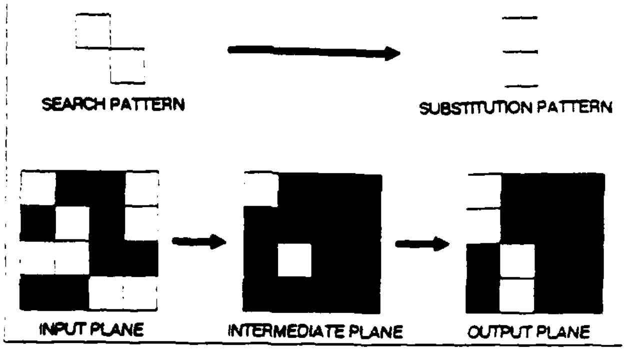


Fig. 9. Pattern substitution.

E. Symbolic Substitution

In addition to parallel optical readout and parallel optical data comparison it is also desirable to include more complex optical processing operations, such as a search with wildcards or a conditioned search before the data are transferred to the electronic system. Certain requirements have to be met, however, by an optical preprocessor for it to be applicable to database systems.

1. *Requirements for an Optical Preprocessor*

In response to a user query, a large number of stored pages are often called up from secondary storage even though generally a large percentage of these pages are not of interest. To reduce the information presented to the electronic system, it is necessary to provide parallel digital optical processing involving memory functions and programmability and to match the processing rate at which data are read from the storage medium. A pipeline architecture is advantageous because the processing rate is more important than the pipeline delay. The length of the pipeline can serve tc adapt the complexity of operation to the requirements.

In addition to this time parallelism provided by the pipeline architecture, spatial parallelism matching the page size on the storage medium is desirable for processing. A typical format consists of 2-Kbyte data pages corresponding to 16 Kbits or a 128 × 128 size pixel array. Operation on these arrays must occur at the readout rate and be rich enough to perform useful work. Finally, the optical processor should not be fixed but programmable to adapt it to various demands.

2. *Principle of Symbolic Substitution*

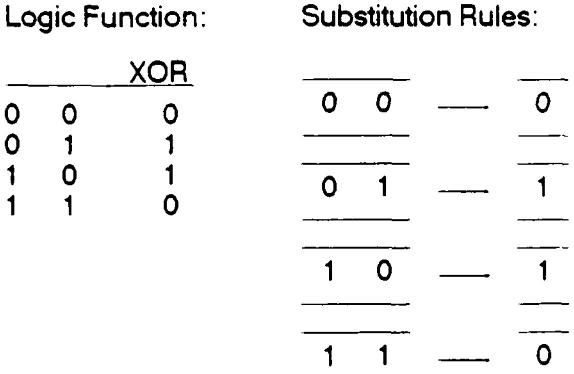
A well-developed technique for optical digital processing is symbolic substitution.[39,40] This logic is able to emulate Boolean logic, cellular logic, arithmetic, and Turing machines. Recently a functionally programmable module was proposed.[41] The elementary operation of symbolic substitution is pattern substitution as indicated in Fig. 9. Each occurrence of the search pattern in the input plane is marked in the intermediate plane by a bright dot. In the substitution phase each of these dots is replaced by the substitution pattern. These primitive operations are easy to implement optically and, with future developments in optical devices, can be executed extremely fast, because the technique can be applied in parallel.

Symbolic substitution operates on binary matrices. Logic can be performed by transforming all the occurrences of a given spatial configuration of binary elements into a different spatial configuration as shown in Fig. 10. Several different transformations can also be implemented optically in parallel. One optical pattern transformation block consists of a recognition part, an optical inverter array, and a substitution part. Both the recognizer and the substituter parts are passive optical components and are matched to their corresponding search pattern and replacement pattern, respectively. The inverter array is the active component, responsible for thresholding and optical power regeneration.

Processing can be achieved by applying several different pattern transformations, also called substitution rules, simultaneously. The parallelism of optics thus is used at a low level to increase speed rather than for high level parallel processing. Although these pattern transformations are global or space invariant operations (the same rules apply to all locations on the array) it has been shown that this mechanism is also able to support local operations.

The time for an N-rule pattern transformation is independent of the number of rules and is given by the propagation time of light through the setup and by the response time of the inverter array. For very fast processing, the propagation time, which could be of the order of 1 ns, could be comparable with the switching time of the inverter array. The progagation time corresponds to the latency of a pipeline processor whereas the throughput depends on the switching time of the inverter array. At high data rates, it is necessary to avoid clock skew. Symbolic substitution supports interconnects with a latency that is constant down to femtoseconds.

Symbolic substitution also supports constant fan in and constant fan out gates, because the substitution rules, specifying the search and the substitution pattern, are fixed. This feature is important because large fan out implies high power consumption and high clock rates can be achieved only if the gates are optimized with respect to a small and constant number of inputs and outputs.



Logic Function:

		XOR
0	0	0
0	1	1
1	0	1
1	1	0

Substitution Rules:

0 0	0
0 1	1
1 0	1
1 1	0

Fig. 10. EXCLUSIVE OR with symbolic substitution.

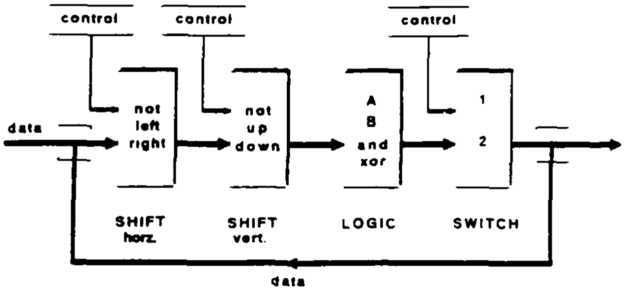


Fig. 11. Functionally programmable module.

Architectually, one substitution rule is implemented by one optical module. Several modules implementing different substitution rules can be arranged either in parallel or in sequence, thus forming an achitecture for a processor. The functionally programmable module[42] consists of a series of transformation blocks to perform controlled shift operations and to perform logic as shown in Fig. 11. Every bit in the array can be programmed to m‹ e in four possible directions. The logic set includes EXCLUSIVE OR, AND, and the identity operator. The program for this module is interlaced with the data and enters as a stream of optical bit arrays.

In an optical parallel pipeline processor, two types of parallelism exist. The first type concerns the parallel processing of many data within a 2-D processing array. This type may be called spatial parallelism. In a pipeline there is also a second type of parallelism. In each stage of the pipeline, an array of data is processed simultaneously, typically by different operations. The degree of parallelism in a pipeline is given by the number of stages. This type of parallelism may be called time parallelism. For a database processor both types of parallelism are applicable. If the processing array is the same size as the readout array, the processing stages have to be cascaded. Between those extremes any trade-off between lateral and longitudinal complexity is feasible.

V. Conclusions

Very large databases and knowledgebases (VLD/KB) are at the heart of existing information systems and will play an even more prominent role in the future. These systems place extreme requirements on existing electronic digital computer technology; requirements that are often not met when VLD/KB are present or real time responses are required.

Optics with its high speed and bandwidth has much to offer for the solution of very large database and knowledgebase problems. In terms of storage, optical disks can hold in order of magnitude more data than magnetic disks per unit area. Although optical disks are currently slower than magnetic disks, the potential exists for at least 2 orders of magnitude greater data rates with multibeam reads. This potential, if realized, would completely change the way database and knowledgebase problems (as well as others) are solved. In addition to optical disks, page-oriented holographic memories hold considerable potential for performance improvement of the solution to these problems. In addition to storing massive components of data, they offer ways to perform processing functions during data retrieval.

The transport of digital data via optical fiber is well developed and its advantages over electronic transport are well recognized. It now appears feasible to remove data from storage and send it through fibers to optical processors without having to convert from photons to electrons. This would have significant performance advantages, especially if data can be read from storage at hundreds of megabytes per second. Such rates would flood current electronic systems since they are designed for magnetic disk rates, which are ~3 Mbytes/s. This leads to new electronic systems as well as new optical systems. Since the data are already in optical form, there are considerable advantages to processing it optically before conversion to electronic form. More research and development of digital optical processors that perform data and knowledge base functions in parallel are needed.

In this paper we have considered many of the ways that optics can play a role in the increase in performance of database and knowledgebase systems. We believe that there is considerable potential for improvement and hope that this paper helps to encourage active research in this important area.

References

1. H. H. Szu and H. J. Caulfield, "Optical Expert Systems," Appl. Opt. **26**, 1943–1947 (1987).
2. D. H. Shin and P. B. Berra, "Computer Archtecture in Logic-Oriented Data/Knowledge Bases," Knowledge Eng. Rev. 4 in press.
3. Latest product literature on the IDM 500 database machine, Britton Lee Corp., Los Gatos, CA (1990).
4. Latest product literature on the DBC 1012 database machine, Teradata Corp., Los Angeles, CA (1990).
5. P. B. Berra and N. Troullinos, "Optical Techniques and Data/Knowledge Base Machines," IEEE Computer **20**, 59–70 (1987).
6. P. B. Berra, A. Ghafoor, P. A. Mitkas, S. J. Marcinkowski, and M. Guizani, "The Impact of Optics on Data and Knowledge Base Systems," IEEE Trans. Knowledge Data Eng. **KDE-1**, 111–132 (1989).
7. D. Z. Anderson, "Coherent Optical Eigenstate Memory," Opt. Lett. **11**, 56–58 (1986).
8. R. A. Athale, H. H. Szu, and C. B. Friedlander, "Optical Implementation of Associative Memory with Controlled Nonlinearity in the Correlation Domain," Opt. Lett. **11**, 482–484 (1986).
9. Y. Owechko, "Optoelectronic Resonator Neural Networks," Appl. Opt. **26**, 5104–5111 (1987).
10. H. J. Caulfield, "Massively Parallel Optical Data Base Management," Proc. Soc. Photo-Opt. Instrum. Eng. **938**, 52–54 (1988).
11. Y. Fainman, H. Rajbenbach, and S. H. Lee, "Applications of Photorefractive Crystals as Basic Computational Modules for Digital Optical Processing," J. Opt. Soc. Am. A *3(13), 16 (1986)*.
12. Y. Fainman and S. H. Lee, "Advances in Applying Nonlinear Optical Crystals to Optical Signal Processing," *Handbook of Signal Processing*, E. H. Chen, Ed. (Marcel Dekker, New York, 1987), Chap. 12.
13. J. E. Rau, "Comparison of Coherently Illuminated Images," J. Opt. Soc. Am. **56**, 541–541 (1966).
14. J. E. Rau, "Detection of Differences in Real Distributions," J. Opt. Soc. Am. **56**, 1490–1494 (1966).
15. C. W. Weaver and J. W. Goodman, "A Technique for Optically Convolving Two Functions," Appl. Opt. **5**, 1248–1250 (1966).
16. F. M. Smits and L. E. Gallaher, Bell Sys. Tech. J. **46**, 1267–1278 (1967).
17. H. J. Caulfield and C. M. Verber, to be submitted.
18. C. Warde, A. M. Weiss, A. D. Fisher, and J. I. Thackara, "Optical Information Processing Characteristics of the Microchannel Spatial Light Modulator," Appi. Opt. **20**, 2066–2074 (1981).
19. C. Warde and J. Thackara, "Operating Modes of the Microchannel Spatial Light Modulators," Opt. Eng. **22**, 695–703 (1983).
20. N. A. Clark and S. T. Lagerwall, "Submicrosecond Bistable Electro-Optic Switching in Liquid Crystals," Appl. Phys. Lett. **36**, 899–901 (1980).
21. N. A. Clark, M. A. Handschy, and S. T. Largerwall, "Ferroelectric Liquid Crystal Electro-Optics Using the Surface Stabilized Structure," Mol. Cryst. Liq. Cryst. **94**, 213–234 (1983).
22. N. A. Clark and S. T. Lagerwall, "Surface-Stabilized Ferroelectric Crystal Electro-Optics: New Multistate Structures and Devices," Ferroelectrics Vol. 59, pp. 69–116 (1984).
23. G. Moddel, K. M. Johnson, W. Li, R. A. Rice, L. A. Pagano-Stauffer, and M. A. Handschy, "High-Speed Binary Optically Addressed Spatial Light Modulator," Appl. Phys. Lett. **55**, 537–539 (1989).
24. D. A. B. Miller, "Quantum Wells for Optical Information Processing," Opt. Eng. **26**, 368–372 (1987).
25. N. Peyqhambarian and H. M. Gibbs, "Optical Bistability for Optical Signal Processing and Computing," Opt. Eng. **24**, 68–73 (1985).
26. K. Y. Han *et al.*, "InP/InGaAs Based Charge-Coupled Devices for MQW Spatial Light Modulator Applications," in *Technical Digest Series* (Optical Society of America, Washington, DC, 1989) Vol. 9, pp. 80–83.
27. W. D. Goodhue, B. E. Burke, B. F. Aul, and K. B. Nichols, "Molecular-Beam Epitaxially Grown Spatial Light Modulators with Charge-Coupled-Device Addressing," J. Vac. Sci. Technol. A **6**, 2356–2360 (1988).

28. S. H. Lee, S. Esener, M. Title, and T. Drabik, "2-D Silicon/PLZT Spatial Light Modulators: Design Considerations and Technology," Opt. Eng. **25**(2): 250–260 (1986).

29. G. R. Knight, "Interface Devices and Memory Materials," *Optical Information Processing Fundamentals*, S. H. Lee, Ed. (Springer-Verlag, New York, 1981), Chap. 4, pp. 158–160.

30. S. Esener and S. H. Lee, "Punch-Through Current Under Diffusion Limited Injection: Analysis and Applications," J. Appl. Phys. **58**, 1380–1387 (1985); "A New Field and Light Controlled Injection Punch-Through Photo-Transistor," paper presented at IEEE Electronics Components Conference, New Orleans (1984).

31. T. H. Lin, M. L. Burgener, S. C. Esener, and S. H. Lee, "Crystallization of Silicon on Electro-Optic PLZT by Laser Beam Modulated in Shape and Intensity Profile," Proc. Mater. Res. Soc. **74**, 135–140 (1987).

32. J. H. Wang, T. H. Lin, S. C. Esener, S. Dasgupta, and S. H. Lee, "NMOS Transistors Fabricated by Simultaneous Laser-Assisted Crystallization and Diffusion on Silicon on Electro-Optic PLZT," Proc. Mater. Res. Soc. **100**, 675 (1988).

33. R. A. Athale and S. H. Lee, "Development of an Optical Parallel Logic Device and a Half-Adder Circuit for Digital Optical Processing," Opt. Eng. **18**, 513–517 (1979).

34. M. A. Handschy, K. M. Johnson, W. T. Cathey, and L. A. Pagano-Stauffer, "Polarization-Based Optical Parallel Logic Gate Utilizing Ferroelectric Liquid Crystals," Opt. Lett. **12**, 611–613 (1987).

35. K. M. Johnson, M. A. Handschy, L. A. Pagano-Stauffer, "Optical Computing and Image Processing with Ferroelectric Liquid Crystals," Opt. Eng. **26**, 385–391 (1987).

36. W. T. Cathey and R. A. Schmidt, "Low Loss Polarization-Based Optical Logic Gates," Proc. Soc. Photo-Opt. Instrum. Eng. **813**, 5–6 (1987).

37. J. Jau, F. Kiamilev, Y. Fainman, S. Esener, and S. H. Lee, "Optical Expert System Based on Matrix-Algebraic Formulation," Appl. Opt. **27**, 5170–5175 (1988).

38. F. Kiamilev, J. Jau, Y. Fainman, S. Esener, and S. H. Lee, "Comparison of Prolog and Matrix-Algebraic AI Systems," J. Opt. Soc. Am. A 4(13), 58 (1987).

39. K.-H. Brenner, A. Huang, and N. Streibl, "Digital Optical Computing with Symbolic Substitution," Appl. Opt. **25**, 3054–3060 (1986).

40. K.-H. Brenner, "New Implementation of Symbolic Substitution Logic," Appl. Opt. **25**, 3061–3064 (1986).

41. K.-H. Brenner, "Programmable Optical Processor Based on Symbolic Substitution," Appl. Opt. **27**, 1687–1691 (1988).

Page Oriented Holographic Memories and Optical Pattern Recognition

H. J. Caulfield
Center for Applied Optics
The University of Alabama in Huntsville
Huntsville, Alabama 35899

Abstract

In the twenty-two years since VanderLugt's introduction of holographic matched filtering, the intensive research carried out throughout the world has led to no applications in complex environment. This leads one to the suspicion that the VanderLugt filter technique is insufficiently complex to handle truly complex problems. Therefore, it is of great interest to increase the complexity of the VanderLugt filtering operation. We introduce here an approach to the real time filter assembly: use of page oriented holographic memories and optically addressed SLMs to achieve intelligent and fast reprogramming of the filters using a 10^4 to 10^6 stored pattern base.

Introduction

Whether the twenty-two years of research on VanderLugt's filtering has been successful or not depends on how one defines success. From a researcher's point of view, it has been very successful. Literally hundreds of Ph.D. theses have been written. Many papers have been written. Simply reviewing the review articles would be a significant task. Therefore, it has been a successful topic in generating research work If, however, success means application of this technology in the field from which most of the money has come: (military applications), then the field has been far from a success. This paper has two goals. First, it seeks to offer an explanation of the apparent failure. Second, it offers a new approach which attacks the problem identified.

Complexity

In numerical calculations, the word complexity has a well-defined meaning. If we regard the operation of pattern recogniton as a well defined numerical operation, we could define the complexity of that operation On the other hand, there are a variety of pattern recognition schemes ranging from correlation with a single prototype to correlation with a large bank of *prototypes to far more complex operations perhaps involving* motion of the mask and/or object. The scene itself has complexity. One measure of this is its information content. This, however is somewhat misleading. If we mean by complexity the difficulty of the problem, the difficulty arises not just from the amount of information that can be packed into a scene not from the within-class and between-class variations of realistic objects. If we model human pattern ognition as a syntactic process with a vast store of rather flexible prototypes, pattern recognition is probably an NP problem.

The point of all this is that realistic problems involve tremendous variations among a vast number of possible prototypes. The idea that one or even a bank of a thousand filter could be adequate to such a task seems, on the surface, highly improbable. There is simply not enough stored information to do the task properly. I believe that this is one of the fundamental reasons VanderLugt filtering has failed to give adequate results for truly realistic complex situations. If this analysis is correct, there is only one possible solution: vastly increase the information available to do the filtering.

Exhaustive Versus Nonexhaustive Search

If we are to store and search a truly vast amount of information, we must reexamine the previous inclination toward exhaustive search of the memory. Clearly, human beings do not employ exhaustive search in their pattern recognition. In reading these words you are searching known patterns of English letters and words using the context of knowing that this is a paper on optical pattern recogniton being written in English and, for some of you, even knowing something of the style of the author. Therefore, you do not have to be searching that part of your memory which deals with the names of your pet dogs or of words in foreign languages or of the map of your city. This represents a compromise between speed and thoroughness. That compromise can be accomplished in many ways. Nevertheless, the important thing for these purposes is to recognize that the compromise was necessary and wise.

The Applications of Page Oriented Holograhic Memories

As is known to a great many of the readers of this paper, page oriented holographic memories allow storitself can be changed. Currently, this is in the region of 1 milisecond per frame, but one microsecond frame time spatial light modulators are currently being constructed at various locations. Thus, at the extreme, we could do an exhaustive search of a million spatial light modulator patterns in the time of one second. Because one second is usually too slow and also because we do not wish to digest that much information each second, it seems prudent to consider intelligent nonexhaustive searching.

Utilization of Stored Information for Pattern Recognition

Because the spatial light modulator is limited to the range of 10^4 to 10^6 pixels, we must be clever in the way we use the stored information. By comparison with a film hologram, for instance, the spatial light modulator may contain only a fraction of a percent of the information content. Two approaches seem reasonable. One is to design pattern recognition kiniforms using yet-to-be-dileanated rules for working in this pixel impoverished environment. The other approach is to store not the filter but the object whose complex conjugated Fourier transform is the filter. We use that "image" to be one input in a joint transform correlator. The balance of this paper is written in such a way as to be independent of that particular choice.

Mask Management

Perhaps the most important feature of the massive memory optical pattern recognizer is the intelligent use of the stored data. That is, the plan for appropriate nonexhaustive search. Because random access to any of the stored patterns can be achieved so rapidly, the physical arrangement of holograms on the common substrate is essentially of no importance. On the other hand, the data base management in the electronic domain that determines which hologram is addressed at any instance needs to be designed very carefully. Thus, as is often the case, it is electronics and not optics that represents the ultimate limitation. We now explore some of the possible data base management concepts which are appropriate for optical pattern recognition.

It may be that what we want to do is arrange the memory in terms of object parameter variations. Those variations may be due to range and/or orientation of the object relative to the observation system. Also stored might be wavelength of those patterns. In any case, there is a multidimensional parameter space which must be searched. This would appear to require a multidimensional tesselation of that space after appropriate scale and distortion of that space to reflect importance and in realistic variability. What is then needed is a lookup table which transforms parameter sets to x-y deflection to call forth the proper mask information from the hologram. Thus we must design a sensible map from the many dimensional space to whatever arbitrary two dimensional pattern we have used to store the data on the hologram. It is not my purpose to discuss the design of multidimensional lookup tables in this paper.

One obvious use of the hologram or holograms is to do exhaustive search by category. The hologram or holograms can be organized in such a way that they are restricted in category or context. For instance in a military environment, one might wish to apply entirely different sets of masks for target acquisition, target tracking and terminal holding. These can represent separate regions on the hologram are even separate holograms. Again, it is not so much the organization of the hologram as the organization of the electronic addresser that is of importance. If the number of contexts or per categories is sufficiently large and sufficient fuzzy, it may be that it is sufficient to specify a context and do exhaustive searching with that context. This represents a two level organization. The first level is to determine the appropriate context. At the second level, we simply do exhaustive searching. From here, it is not hard to generalize to a multi level search. Broad context are sought and then narrow context sought within those. This establishes a tree structure.

It is possible to consider composite masks or as variously termed "linear combinations of matched filters" (1) or "composite matched filters" (2). We accomplish positive weightings by sequencing through all of the positively weighted components and varying, for example, the intensity of the laser beam. The time integrated correlation plane pattern is then stored. Next, we generate the sum of negatively weighted components in a second memory. Finally, we sub stract the two integrated images to obtain the desired results. We then have the potentiality of generating quite general filters simply by controlling the waitings. Implicit in this is the assumption that we can run through a wide variety of masks very rapidly. Even with the slowest of current spatial light modulators, it takes only two frames to do a general composite matched filtering. That is because we are averaging or integrating during the entire cycle. The weights can be predetermined or even adaptively determined.

Conclusion

The material just presented is an outline of one approach to the vast increase in complexity that is probably needed to make optical pattern recognition practical for many purposes. Because the page oriented holographic offers storage and access capabilities far beyond those which can be offered electronically, the value of optics is enhanced. That is, this is a clear illustration of a case in which optics can make practical what would be essentially impossible electronically. As with any new solution, this one carries with it a great many new problems. I hope that the outline of these problems is greeted as an opportunity for invention and not as an excuse for inaction. Any time we have an opportuity to do something important that cannot be done electronically, we should explore that opportunity carefully.

References

1. H. J. Caulfield, R. Haimes, J. Horner, "Composite Matched Filters," Gabor Memorial Issue, Israel Journal Of Technol. Vol. 18, 263, (1980).

2. H. J. Caulfield, "Linear Combinations of Filters for Character Recognition: A Unified Treatment," Applied Opt. Vol. 19, 3877 (1981).

PAGE ORIENTED HOLOGRAPHIC MEMORY ADDRESSING OF OPTICAL BISTABLE DEVICES ARRAYS

H. J. Caulfield

Center for Applied Optics, The University of Alabama in Huntsville
Huntsville, Alabama 35899

Abstract

Page Oriented Holographic Memories can be used as stored microprograms. Because they may not give highly accurate signal levels, e.g. to 0.1% as may be needed, these holograms should address not a device array but an array of optical bistable shutters through which adjustable stable light beams may pass.

Page Oriented Holographic Memories

Their Glorious Past

There was a time in the mid 1960's and 1970's when page oriented holographic memories (hereafter, POHM's) held center stage in the world's holography effort. The United States efforts included massive efforts by IBM (Lohmann, Bryngdahl, etc.) and AT&T (Collier, Borckhardt, Lin, Anderson, etc.) The most massive European effort was by Philips. The hope was to build a read-write memory of gigantic storage capacity (say, 10^{10} bits) with exceptional random access time (say, 10^{-8} seconds). This 10^{18} or better bit per second random access memory was to fee the hoped for supercomputer of the 1980s.

The basic concept (1) is extremely simple. The first Figure illustrates it schematically. Many small holograms were to be formed on nonoverlapping areas of the same substrate. A 1mm diameter subhologram might produce an image of a 1024 x 1024 array of on - off points. The real image from each subhologram is formed at the same place in space. At that place an array of parallel read out detectors was to be placed. By deflecting a laser beam so that the proper hologram was illuminated, we could cause any one of the stored point patterns to hit the parallel detector array essentially instantly. Deflectors of nanosecond random access capabilities were built. Careers were built. Soon, however, the projects were abandoned.

What went wrong? In a word: everything. The search for a suitable read-write material failed. The ultimate use failed to materialize as did arrays of 1024 x 1024 parallel readout 1 nanosecond detectors. It was never clear how supercomputers could use that much data at that rate.

Their Inglorious Present

Across the world, POHMs are dead. It is not even a serious research field. The sole exception to this sad tale is the Soviet Union which has what appears to be a significant effort in this field. Someone, we or they, is wrong.

Why are the Soviets doing this? If we only read their English language publications (2), the answer becomes clear.

Their Glorious Future

The future of POHMs is optical computing. We use the POHM to reprogram an optically addressed spatial light modulator (SLM). SLMs are, at last, becoming fast. Parallel read in and parallel read out are trivial. Furthermore, for many purposes, a read only POHM suffices. Every objection to the POHM from the 1960s has vanished in the 1980s.

Uses With Optical Bistable Device (OBD) Arrays

Enabling

We may wish to use the OBD array for blocking unwanted interconnections. For instance an optical crossbar could be formed using an N x N OBD array to perform 1 to 1 connection of N sources to N receivers (3). For each of the N^2 possible interconnections there is precisely one on-off pattern for the OBD array that achieves it. The signal beams themselves can be relatively weak so that even N of them will not switch the OBD. A POHM could then be used to switch (enable) the appropriate OBDs. Wavelength, polarization, angle, or some combination of these can separate the signal beam from the stronger enabling beam. Parallel addressing makes the reprogramming fast (limited by either the OBD or the light deflector)

Optical Fredkin Gate Arrays

In a prior publication (4) we showed that optical Fredkin gate arrays allow

- logic (all functions),
- memory, and
- interconnections.

A pending publication (5) shows that N(N-1)/2 Fredkin gate array can make an N x N optical crossbar switch If the array can be addressed optically it can be switched by a POHM. All of this is of interest to an OBD conference only if OBD Fredkin gates are possible. We now show that they are.

An ordinary logical gate is really a nonlinear function generator. Two input binary variables (A and B) generate a binary output C. Since there are only four possible A, B patterns, C is often represented by a truth table. The truth table for the AND function is shown below.

A	B	C = A AND B
0	0	0
0	1	1
1	0	1
1	1	1

There are fewer outputs than inputs, so information is lost. For example, if A AND B is 1, we can no longer say what values A and B had.

A Fredkin gate conserves information. The next Figure shows a Fredkin gate schematically. There are three inputs (A, B, and C) and three outputs (A', B', and C'). Given one set of three, we can infer the other set using

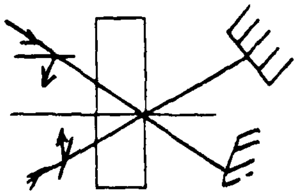
C' ← C

IF
C=0
A' ← A
B' ← B

IF
C=1
A' ← B
B' ← A

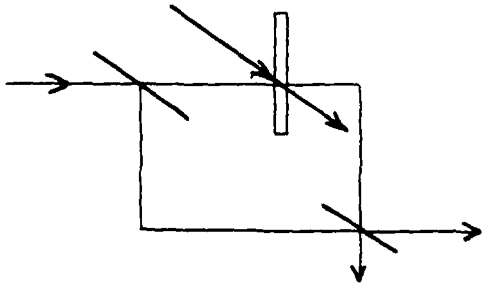
Reference 5 and other references therein show that *such gates can perform* all logical functions, many memory operations, and quite generalized switching.

We turn now to OBD Fredkin gates. One way to assemble one of these is shown below.



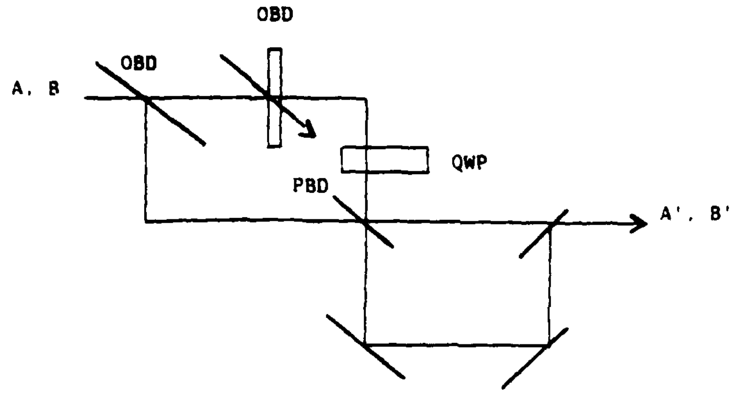
The intensities of A and B are below threshold and so is their sum. so with C=0, the OBD reflects achieving A'=A and B'=B. With sufficient applied signal (C=1), the OBD transmits giving A'=B and B'=A.

A second version is shown in Figure below. The A states is vertically polarized. The B state is vertically polarized. An ordinary



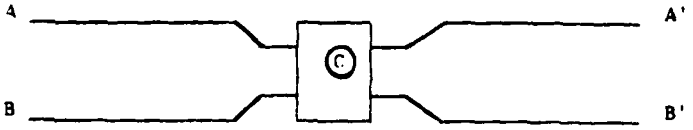
beamsplitter (OBS) directs light to a polarizing beamsplitter (PBS) which directs the vertically polarized light down to form A' and transmits the horizontally polarized light to form B'. Thus in the reflective mode of the OBM (C=0), A'=A and B'=B. When the OBD threshold is exceeded (C=1), the vertically polarized light A, is reflected into the B' channel and vice versa.

A third version, preferable to the second for cascading, marks A and A' with vertical polarization and B and B' with vertical polarization. This is easy to do as shown in Figure below.

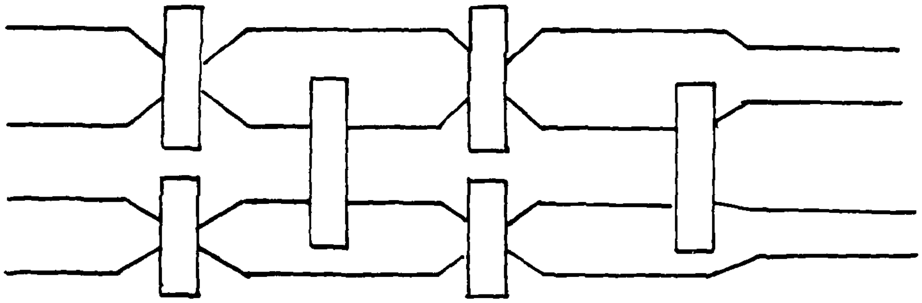


Also shown in Figure above is a lossless recombiner.

Let us represent a generic OBD Fredkin gate as in Figure below.



We can then combine these in various ways (6). For example, to connect a linear array of 2^N sources to a linear array of 2^N detectors we need 2N layers of OBD Fredkin gates as indicated in Figure below for N=2.



If we use a POHM to switch the N(N-1)/2 OBD Fredkin gates, the switching time is limited by the slower of two times. the detector response time or the laser deflection time.

CONCLUSIONS

Optical bistable devices (OBDs) can be viewed as optically controllable operators. Arrays of optical bistable devices can be "programmed" by patterned light from any of 10^4 to 10^6 holograms any of which can be accessed in a laser deflection time (from milliseconds mechanically to microseconds acoustooptically to to nanoseconds electrooptically). If the OBDs can respond in nanoseconds, this represents the switching speed. Although 10^4 to 10^6 is a large number of "programs," it is certainly finite. In this sense we have a Reduced Instruction Set Computer

1. F. M. Smits, and L. E. Gallaher. (1967). Bell Syst. Tech. J. 46, 1267-1278.

2. H. Elion, V. N. Morozov, "Optoelectronic Switching Systems in Telecommunications and Computers," Marcell Dekker, 1984.

3. J. W. Goodman, A. R. Dias, and L. M. Woody, "Fully Parallel, High-Speed Incoherent Optical Method for Performing Discrete Fourier Transforms," Opt. Lett. 2, 1 (1978).

4. H. J. Caulfield, W. Miceli, R. J. Seymour and J. Shamir, "Optical Computing and the Fredkin Gates," Appl. Opt. Vol. 25, No. 10, pg. 1604-1607 (May, 1986.)

5. J. Shamir, H. J. Caulfield, "High-Efficiency Rapidly Programmable Optical Interconnections," accepted to be published in Applied Optics, March, 1986.

AUTHOR INDEX

Invited Paper

OPTICAL MAPPING APPLICATIONS

H. J. Caulfield

The University of Alabama in Huntsville, Center for Applied Optics
Huntsville, Alabama 35899

ABSTRACT

Holography can be used for arbitrary, parallel, weighted mappings between planes with up to to 10^6 pixels each at I/O limited rates. This allows precalculated mappings to occur at very high speed. The applications for one-to-one, one-to-many, many-to-one, and many-to-many maps are explored here.

1. INTRODUCTION

Recently (1,2), I have shown that it is possible to map large .nput scenes (up to to 1000 x 1000) into large output scenes (up to 1000 x 1000) using arrays of holograms. Figure 1 shows a schematic drawing of a passive (Spatial Light Modulator or SLM) input system. Figure 2 is a schematic drawing of an active (source array) input. We will devote less attention to the hardware than to the applications in what follows.

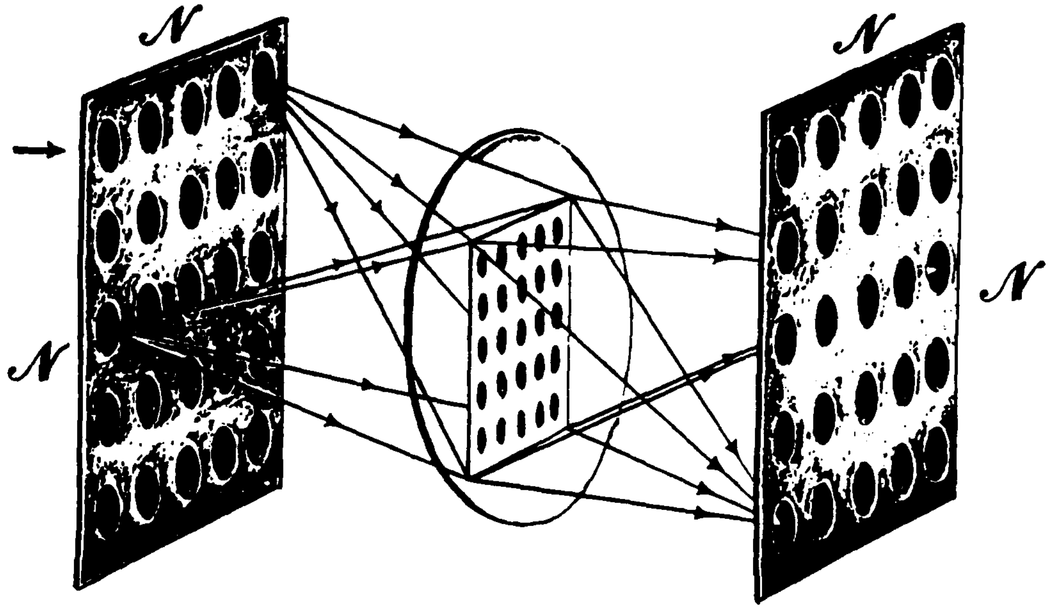


Fig. 1. An NXN hologram array is imaged onto an NXN output array through an SLM. Each hologram illuminates the SLM with a unique pattern of light. All work in parallel.

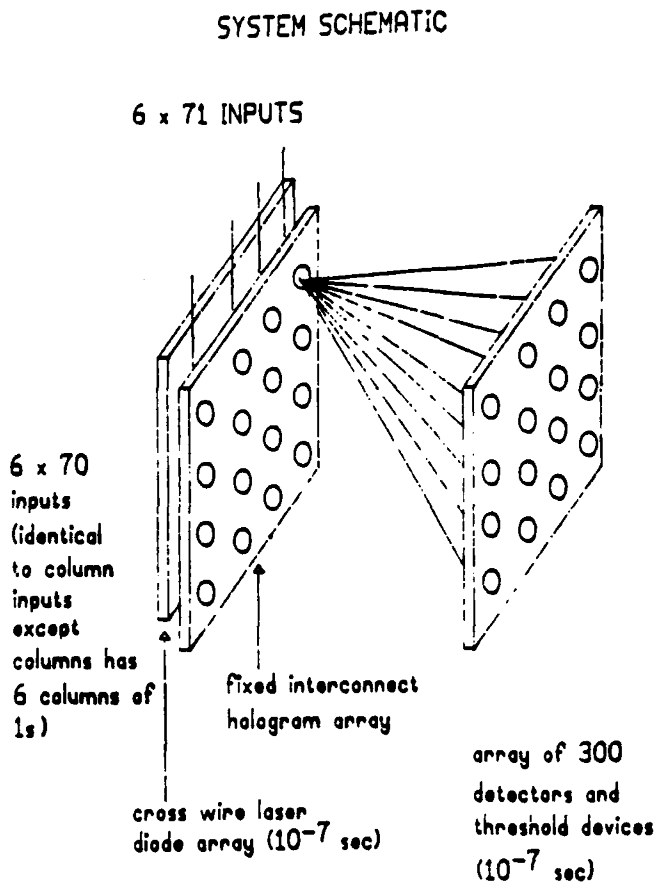

SYSTEM SCHEMATIC
6 x 71 INPUTS
6 x 70 inputs (identical to column inputs except columns has 6 columns of 1s)
fixed interconnect hologram array
cross wire laser diode array (10^{-7} sec)
array of 300 detectors and threshold devices (10^{-7} sec)

Fig. 2. An Active Massive Interconnect System Using An Array Of Modulated Sources As The Input.

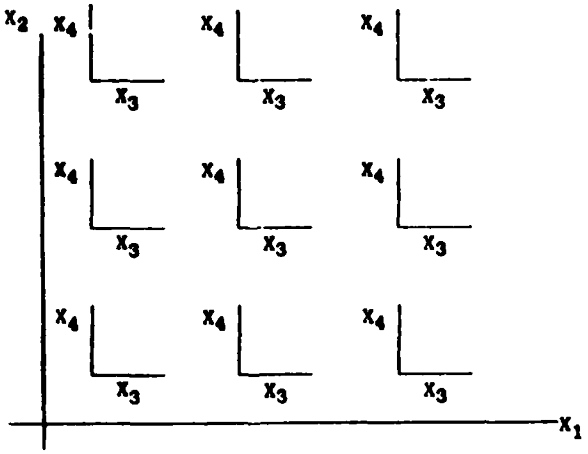


Fig. 3. One Way of Representing A Four-Dimensional Space in Two Dimensions.

The three types of mapping of interest below are one-to-one, one-to-many, and many-to-many.

2. ONE-TO-ONE MAPPING

This applications is sometimes called "coordinate transformation." Some familiar transformations are

$$\left.\begin{array}{l} x,y \rightarrow r,\theta \\ r,\theta \rightarrow x,y \end{array}\right\} \text{Polar}$$

$$x,y \rightarrow \log(x),\log(y)$$
$$x,y \rightarrow \exp(x),\exp(y)$$

etc.

The patterns are calculated and embodied in holograms. The transformations are then speed limited by Input/Output or I/O.

It is not necessary that the mapping have a geometric interpretation. For example, x and y could be the first two principal components in an N dimensional feature space (the two orthogonal vectors in that space along which best separation among events occurs). Thus the x-y location of an event gives a closest-category interpretation an a probability measure. We can use as an output one dimension along which categories are arrayed and a second dimension which gives the probability scale for that event.

Likewise mappings need not be confined to two dimensions. We can represent N dimensional spaces by positions along a one-dimensional space filling curve. Or we can sample the space in some more pictorial fashion such as shown in Figure 3.

In principle, we do not absolutely require uniform spacing in either the input plane or the output plane, so, for example, uniformly spaced x, y points can be transformed into nonuniformly spaced γ,θ points. This allows an "exact" (no interpolation/extrapolation) mapping. This can present an accuracy problem if it requires the holograms to overlap.

3. ONE TO-MANY AND MANY-TO-ONE MAPPINGS

A good example of the use of holographic many-to-one mapping is in an optical Dempster-Shafer (D-S) evidential reasoning machine (3). It is easy to show that to update our beliefs on the basis of new evidence using vector outer products to "correlate" evidence and holograms to route

- outer product terms consistent with proposition P_i to a detector to give the unnormalized support S_i outer product terms inconsistent with P_i to a detector to give the unnormalized doubt D_i, and

- mutually inconsistent outer product terms into a single detector to give a term I.

We then calculate our new beliefs

$$b_i = (S_i, P_i)$$

about proposition P_i, where

$$S_i = \text{support for } P_i = S_i/(1-I)$$

and

$$P_i = \text{plausibility of } P_i = [1-(D_i/(1-I))].$$

A good one-to-many application is the Hough transform. In a Hough transform for parametric fit of straight lines to x,y input points, we might use equations of the form

$$y = mx+b.$$

The straight lines through the point x_0y_0 satisfy

$$b = -x_0m+y_0.$$

That is a point in the x,y plane maps into a straight line in b,m space. Two points map into two straight lines. The intersection of those lines gives the b and m of the line through both points. Many x,y points lead to many intersections in b,m. On the other hand, points "pile up" near b,m points which represent multiple point straight lines in x,y. With this method we can do large Hough transforms in O(1) time (~ milliseconds due to I/O).

4. MANY-TO-MANY MAPPINGS

There is evidence that biological reasoning uses mapping of information from one "frame of reference" to another (4). Locations representing multiple data coincidence neighborhoods can map into decision plane neighborhoods in another plane.

Of course, the ultimate computer goal is intelligence. The physical basis for an intelligent computer must be highly complex. To achieve useful intelligence, we will need high speed as well. Using this method we can interconnect each of a 1000 x 1000 input array to each of a 1000 x 1000 output array fully in parallel. This combination of complexity and speed (e.g. 10^{12} interconnections in a millisecond or 10^{15} connections per second) could serve as the physical basis for true intelligence if much more attention is devoted to how to transduce cognitive concepts into the appropriate form of entry into this system ( ).

5. CONCLUSION

The ability of optics to perform full interconnection from a large input array to a large output array in parallel creates many new possibilities. Fast algorithms, e.g. for Hough transforms, are not needed. If we need the speed and can afford the hardware, they can be done in the time required for I/O. Advanced computational methods which would be too slow with partially serial electronics become feasible with parallel optics. In particular massive neural networks fit in this category.

6. REFERENCES

1. H. John Caulfield, "Parallel N^4 Weighted Optical Interconnections." Applied Optics, Vol. 26, No. 19, October, 1987.

2. H. J. Caulfield, R. Barry Johnson, J. Shamir, "Practical Considerations for Massive Holographic Interconnection." In preparation for Applied Optics

3. G. Shafer, "A Mathematical Theory of Evidance." Princeton University Press, Princeton, NJ (1976).

4. Jack J. Gelfand and Richard M. Peterson, "A Computational Map Approach to Logic Processing." SRI Labs, Private Communication, 1987.

7. ACKNOWLEDGEMENT

This work was supported in part by work done on Contract #N00014-86-K-0591, Dept. of Navy, Office of Chief of Naval Research.

STACKED PAGE ORIENTED HOLOGRAPHIC MEMORY

H. J. Caulfield

The University of Alabama in Huntsville, Center for Applied Optics
Huntsville, Alabama 35899

ABSTRACT

While page oriented holographic memories are extremely valuable, they can take a great amount of lateral space. We show here how to stack a number of holograms in such a way that we can select one layer to be "active." As a result, the lateral area needed to store a given number of holograms is reduced by L, the

1. INTRODUCTION

Conceived of by Smits and Gallaher (1) as a computer memory, the Page Oriented Holographic Memory (POHM) appears to have found many applications in optical computing (2,3,4,5,6). The basic POHM geometry is shown in Fig. 2. A laser beam is deflected to the proper subhologram. Whatever hologram is illuminated produces an output light pattern at a preselected location. In optical computing, we normally use an optically addressed spatial light modulator at that location.

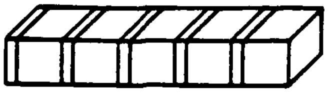


FIG. 2. LIGHT ENTERING FROM THE LEFT CAN BE DIRECTED OUT ANY OF THE CELLS.

The primary problem with POHMs is that the subholograms need to be one to two millimeters in diameter. If we want to have, say, a 1000 x 1000 array; we need at least one square meter of substrate.

The goal of this work is to find a way to compact the POHM laterally by extending it longitudinally into L layers. If we can then select the layer of interest, we can address by x, y, and k, where k is the index

2. LAYER SELECTION

Polarization seems to be the most logical layer selection method. A longitudinally Pockels cell can change the polarization of the light passing through it. A second longitudinally Pockels cell can change the polarization back to its original state. Thus, what we want is a POHM which works for one polarization state but not for the orthogonal state. For many years I have sought suitably asymmetric holograms. The only hologram with a truly massive asymmetry I have come up with (thanks to Steve Case and Tomascz Jannson) is a thick hologram in which the rays are diffracted by 90° inside the hologram. For a variety of reasons, this is not a good solution for stacked POHMs. Thus I turned to what I call "polarization transducers" - devices which convert polarization into other properties.

A polarization transducer is a device that changes polarization (which is easy to control) into some other property which might be more difficult to control. In particular, having had some previous experience in using polarization switches and birefringent prisms to direct light to suitable holograms (2), I thought to apply this technology to stacked holograms. What now follows is a step by step description of the buildup of a stacked hologram array.

First, we can consider a single element as shown in Figure 1. Depending on the polarization state, which depends on the incidence polarization and whether the switch is on or off, light is either transmitted through the prism or deflected downward. The light which continues to propagate can enter other such units. The light deflected downward strikes a hologram on the bottom side of the prism. Thus, that hologram is eliminated or not eliminated depending on whether the polarization of light passing through the switch is proper.

Second, we can stack a large number of these longitudinally. Figure 2 shows the scheme. Clearly, one addressing beam can address any of those holograms.

Third, we can arrange a plane filled with such devices as shown in Figure 3. Now, wherever we put a beam along a line, we can readout from a particular layer. That is, we can address a two dimensional array of holograms via a one dimensional scan plus polarization switching of layers.

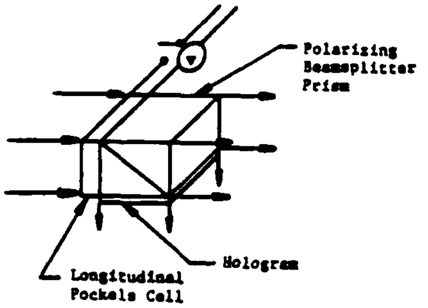


Fig. 1. Light is incident of a polarization switch which either rotates the plane of polarization 90° or leaves it unrotated. The light then enters a polarizing prism. Depending on the polarization, that prism transmits or deflects the beam downward by 90°. A hologram on the bottom of the prism can, therefore, be addressed. We assume that the hologram is highly efficient, transmitting at most 10^{-3} of the incident light. Such holograms are now routine in dichromated gelatin.

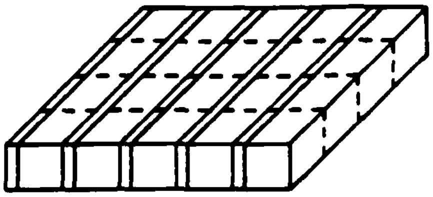


FIG. 3. BY EXPANDING THE PRISMS AND CELLS CONTINUOUSLY WE CAN MAKE A SHEET OF CELLS OF THE FIG. 1 TYPE. THE DIVISIONS, SHOWN DASHED, CORRESPOND TO SEPARATE LIGHT PATHS BUT NOT TO PHYSICAL DIVISIONS.

Fourth, we come to the most difficult part of stacking these layers on top of each so that we fill a three dimensional space full of accessible holograms (Fig. 4). The problem with this scheme is that each layer must be readout through all layers between it and the target plane. Those layers contain a variety of switches, prisms and holograms. It remains to be seen what quality of image can be formed through these. Also polarization effects could be disastrous. What is certain, is that the exposures must take place through all intervening media. Holograms so recorded can compensate for a great many minor refractive defects. One the other hand, they cannot compensate for changes in polarization or for light bent away from the region of the hologram itself. In all likelihood, it will take considerable experimentation to learn to do this properly.

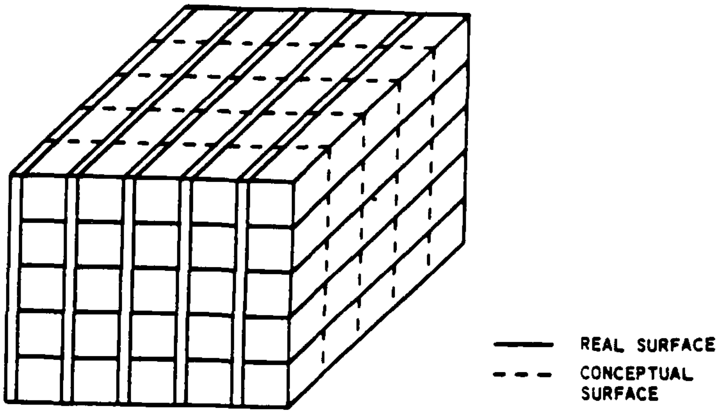


FIG. 4. IN THE 3D VERSION, THE POCKELS CELLS ARE CONTINUOUS IN TWO DIMENSIONS WHILE THE PRISMS FALL INTO DISTINCT LAYERS BUT ARE CONTINUOUS WITHIN THE LAYERS.

3. DISCUSSION

The primary effect of the scheme we have just discussed is to fill a three dimensional space rather than a two dimensional space with page oriented holograms. This is clearly a better use of space than the traditional POHM. On the other hand, much remains to be explored concerning how well holograms can perform through layers of other holograms. Some sort of quasi Fourier transform hologram seems indicated, see Fig. 5. To select a hologram, we select a layer by switching one of the Pockels cells and then deflect x-y position.

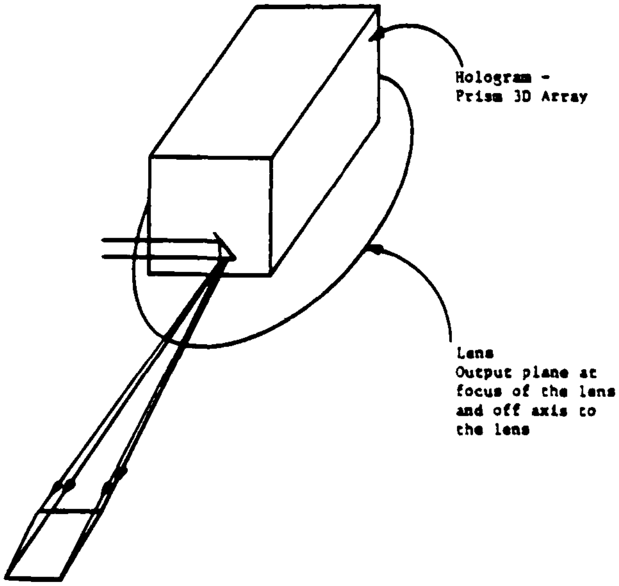

Hologram - Prism 3D Array
Lens
Output plane at focus of the lens and off axis to the lens

FIG. 5. EACH HOLOGRAM CAN BE A FOURIER TRANSFORM HOLOGRAM, SO THERE IS A COMMON OUTPUT PLANE. THAT PLANE IS OFF AXIS SO EFFICIENT HOLOGRAMS CAN BE · ADE. AN EXAMPLE HOLOGRAM SELECTION IS SHOWN.

4. REFERENCES

1. F. M. Smits and L. E. Gallaher, "Design Considerations for a Semipermanent Optical Memory," Bell Syst. Tech. J. 46, 1267 (1967).

2. H. J. Caulfield and H. H. Szu, "Optical Expert Systems," Applied Optics, Vol. 26, No. 10, pp 1943-1947, May (1987)

3. H. J. Caulfield, "Parallel N^4 Weighted Optical Interconnections," Applied Optics, Vol. 26, No. 19, Oct (1987).

4. M. M. Mirsalehi, J. Shamir, and H. J. Caulfield, "Residue Arithmetic Processing Utilizing Optical Fredkin Gate Arrays," Applied Optics, Vol. 26, No. 18, pp 3940-3946, Sept. (1987).

5. H. J. Caulfield, "Page Oriented Holographic Memories and Expert Optical Systems," Proceedings of the SPIE, Vol. 752, pp 172-174, Digital Optical Computing (1987).

6. H. J. Caulfield, "Page Oriented Holographic Memories and Optical Pattern Recognition," Proceedings of the SPIE, Vol. 754, pp 74-76, Optical and Digital Pattern Recognition (1987).

7. B. McManus and H. J. Caulfield, Applied Optics (to be published).

APPENDIX D

MISCELLANEOUS APPLICATIONS

1. FREDKIN GATES

The initial papers in this field (Appl. Opt. 25, 1604 and SPIE 625, 2) stirred much research and is widely cited. It has led (not sponsored by ONR) to sub kT operation of optical processors. Our ONR sponsored work was on various applications such as rapidly programmable switches (Appl. Opt. 26, 1032), new configurations (Appl. Opt 26, 3455), and a more compact residue arithmetic architecture (Appl. Opt. 26, 3940.

2. PATTERN RECOGNITION

Despite the wonderful invariance properties of prior recognition masks, they were very difficult to manufacture. We showed how to simplify mask design and manufacture tremendously (Appl. Opt. 26, 2311; SPIE 613, 260; and Appl. Opt. 27, 2895). This work has led to much other work (still ongoing under other sponsorship at several institutions).

3. NEURAL NETWORKS

While neural networks are the obvious application of massively parallel optical interconnections, they present huge accuracy problems. We showed the first general way to train neural networks for low accuracy operation (IEEE Trans. Systems, Man, and Cybernetics, accepted). We then mapped out a general approach to utilizing the new-found complexity capability (WNN-AIND 90 AND IJCNN 90).

4. FUNDAMENTAL BOUNDS

One of the most cited papers from this whole contract is the demonstration that parallel optical processors have a fundamental speed limit of about 0.01 GH^3 (Appl. Opt. 26, 1567).

Optical computing and the Fredkin gates

Joseph Shamir, H. John Caulfield, William Micelli, and Robert J. Seymour

The use of optics to implement the Boolean logic functions traditionally used in conventional electronic computing is an active area of optical computing research. Many proposed optical implementations duplicate the configuration of electronic logic gates and hence may not optimally utilize the full benefits of optical techniques. We present here a new optical gate, the Fredkin gate, which may, in principle, be minimally dissipative (i.e., exhibit reversible logic) and whose response time may be limited in some implementations only by the duration of optical pulses (i.e., in the picosecond range). Such gates, which consist of three input and three output lines, can be programmed to produce a standard set of Boolean functions and appear well matched to the parallelism of optics. We present here a number of optical implementations of Fredkin gates and suggest ways of composing their interconnections to achieve combinatorial logic, circulating memories and generalized interconnects.

I. Introduction

"The energy requirements of basic logic operations ultimately impose fundamental limits on achievable computation rates and all largely independent of device implementation technology." [1] Part of this energy consumption is due to the intrinsic nature of the traditional composition of logic elements. This fact becomes evident if we recall that a conventional logic gate has more input lines than output lines. Thus some of the information coming into the gate is lost and cannot be retrieved. The irreversible nature of the gate makes it dissipative not only in information but also in energy. In an effort to overcome these limitations, Fredkin and Toffoli[2] proposed a new kind of logic gate which has the same number of output lines as it has input lines. Fredkin gates are capable of performing conventional logic operations while preserving all the original information. In contrast to the conventional logic gates the Fredkin gates may, in principle, be run backward to regenerate the original input signals.

The purpose of this work is to introduce the optical Fredkin gate, illustrate its programmability, and suggest it as a basic building block of an optical computer. An overview of the main aspects of the Fredkin gate is given in the next section followed by a variety of proposed optical implementations. A number of useful configurations are discussed in a final section.

II. Background of the Fredkin Gates

The basic Fredkin gate is defined as a black box having three binary inputs and three binary outputs (Fig. 1). The C-input, the control line, determines the operations of the gate on the other two inputs according to the following rules:

$$\begin{aligned} & C = C' \\ \text{if} \quad & C = 0\colon\ A' = A; \quad B' = B; \\ \text{if} \quad & C = 1\colon\ A' = B; \quad B' = A. \end{aligned} \tag{1}$$

It is quite evident that this gate is reversible; i.e., it may be run backward to return to the original inputs, and, therefore, it is in principle nondissipative. [The original definitions used in Ref. 2 are the inverse of Eq. (1); however, we find this definition more intuitive and more suitable for optical implementation.]

Using the three inputs and the three outputs of the Fredkin gate, one may implement the traditional logic gates that usually have two input lines and one output line. To make the comparison easier, in the examples of Fig. 2 we leave the lines corresponding to the conventional gates straight while the other lines are shown bent. In Fig. 2(a) an AND gate is implemented keeping the a input at the 0 level and obtaining the required output of the A' line. Unlike conventional gates, we obtain two additional outputs that we may utilize or ignore. In a similar way, one possible implementation of an OR gate is shown in Fig. 2(b). It can be easily shown that any other function, such as NOT, FAN-OUT, FAN-IN, and FLIP-FLOPs, is also easily implemented.

Robert Seymour is with GTE Laboratories, Inc., 40 Sylvan Road, Waltham, Massachusetts 02254; W. Micelli is with Naval Ocean Systems Center, San Diego, California 92152; the other authors are with University of Alabama in Huntsville, Center for Applied Optics, Huntsville, Alabama 35899.

Received 1 October 1985.
0003-6935/86/101604-04$02.00/0.

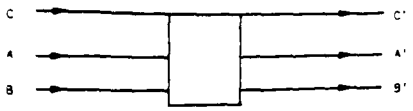


Fig. 1. Fredkin gate (see also text).

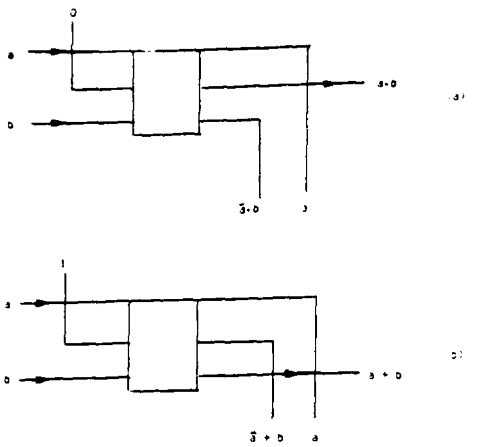


Fig. 2. Logic gates implemented by the use of Fredkin gates: (a) an AND gate; (b) OR gate.

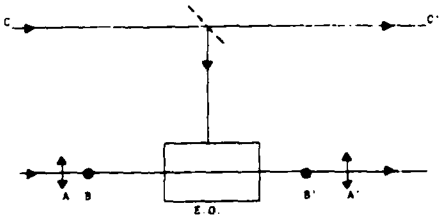


Fig. 3. Polarization switching gate.

In the next section we discuss a number of ways to implement the Fredkin gate by optical and electrooptical means.

III. Optical Implementations of the Fredkin Gate

For applications in logic networks one is usually interested in logic gates containing nonlinear bistable elements. The basic configuration of a Fredkin gate, however, is not restricted to digital signals, and, in principle, one may use these gates for processing analog signals as well. In the examples that follow the nature of the control signal will determine the actual response of the gate.

A. Polarization Switching Gate

A polarization switching gate is shown in Fig. 3. The *a* and *b* lines correspond to two orthogonal polarizations of a light beam (or a waveguide channel of an integrated optical system) traversing an electrooptic modulator that rotates both polarizations by 90° when activated. The activation is induced by the *C*-line either by a direct electronic pulse or, as shown in the figure, by an optical signal transduced to an electronic signal using a photodetector (photoconductor or photodiode–amplifier combination). Polarizing beam splitters may be applied whenever a spatial separation is required between the *A* and *B* lines. The main advantage of this gate is its relative simplicity, while its disadvantage is the different nature of the *C*-line that also changes level during transition through a gate (i.e., there is a lower light intensity in *C*′ than in *C*; this effect may, however, be corrected by incorporating an amplifying medium on the line).

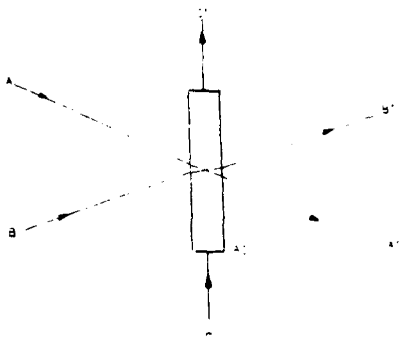


Fig 4. Acoustooptic gate.

B. Acoustooptic Gate

In Fig. 4 we show a schematic diagram of the acoustooptic gate: The two input lines are laser beams incident on an acoustooptic deflector (either bulk or integrated SAW) at the Bragg angle. If there is no acoustic signal ($C = 0$), the two beams continue unaffected (A' and B'), while if C is present each beam is deflected into the other channel. This is also a simple gate, but here too one has a *C*-line which is basically different in nature than the other two lines. Nevertheless, this kind of gate can be easily cascaded and integrated. For example, a single acoustic pulse may activate many gates as it travels along the system.

Of course, any 100% efficient gateable diffractor will suffice. Such devices are possible in integrated optics.

C. Photorefractive Gate

The photorefractive gate based on four-wave mixing is an all optical gate with one of its tentative implementations illustrated in Fig. 5. In this case the *C*-line constitutes the two pump beams. The inputs *A* and *B* are transmitted if *C* is absent and phase-conjugated when the pump is present resulting in switching between the outputs.

D. Waveguide or Coupler

In optical communication and integrated optical systems a modulated waveguide or fiber coupler may serve as a Fredkin gate. Two general classes of this kind of gate may be implemented. The out-of-plane control is shown schematically in Fig. 6(a) and the in-plane control with one possibility depicted is Fig. 6(b). A number of workers have already implemented the electronically addressed coupler[4,5] that may serve as a Fredkin gate with an electronic *C*-input. To symmetrize the system one may use photodetection combined with the electrooptic coupler to *facilitate optical* control. A more advanced technology would be the

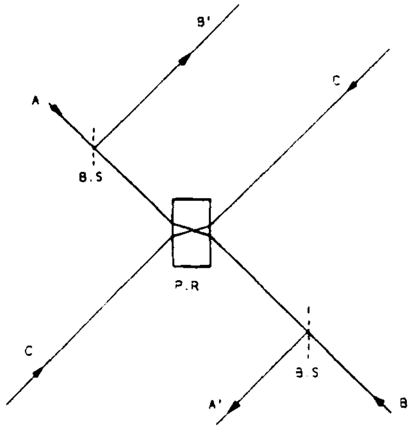


Fig. 5. Photorefractive gate using four-wave mixing in photorefractive material (*P.R*). Beam splitters (*B.S*) are needed for output coupling.

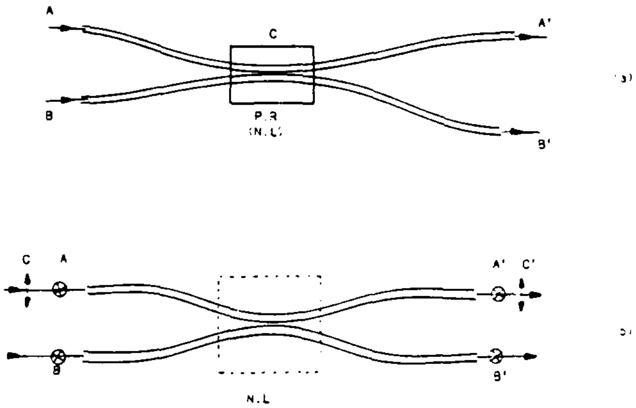


Fig. 6. Waveguide coupler gate. The coupling region activated by line *C* is a photorefractive or other nonlinear material or electrooptic material: (a) out-of-plane control; (b) in-plane control.

use of photorefractive material for direct optical control of the coupling constant. The example in Fig. 6(b) is a waveguide coupler incorporating highly anisotropic guides containing nonlinear material. The two coupling waves (*A* and *B*) are ˈntroduced with the same polarization so that they can couple while the control signal *C* is orthogonally polarized so that its power is used to activate the coupling between the *A* and *B* channels, but it does not couple itself into the other guide.[6]

IV. Proposed Devices Incorporating Fredkin Gates

We demonstrate the applicability of these new gates by proposing, in addition to the conventional logic gates, two very useful devices that incorporate arrays of the waveguide gates shown in Fig. 6.

A. Optical Crossbar

The gate array of Fig. 7 may be constructed of gates of the type depicted in Fig. 6(a) or the type in Fig. 6(b). In the first case each gate may be accessed randomly from above by an electric field or by light, depending on the specific device used. As we are dealing with optical computing we might prefer activation by light such as a holographic coupler[8] or fiber coupler. With proper addressing each input line can be coupled to each output line. This system may prove to be an extremely fast and efficient crossbar or optical switch board. The in-plane addressing of Fig. 6(b) is applicable if one desires to activate a whole column together. At first sight it appears that this kind of addressing is not suitable for random access; however, with very fast pulses this also becomes feasible.

B. Tapped Delay Line

The basic configuration of Fig. 8(a) is a tapped delay line. A fiber ring may be utilized for long delays, while for very short delays one may use waveguide rings, the feasibility of which has also been demonstrated.[9,1] Here too the addressing may be of the first type [Fig. 6(a)] or of the second type [Fig. 6(b)]. Such a setup may be used to delay all the energy in a pulse or just part of it to produce a pulse train from a single initial pulse. A slight modification of the system as illustrated in Fig. 8(b) may be used to reverse the direction of signal flow resulting in a true reversible Fredkin gate. In the future, an optical memory block may resemble the array depicted in Fig. 8(c). This seems to be a short-term memory, but with the integration of amplifying medium it may serve also as a long-term memory.

V. Discussion

Initial approaches to optical computing have tended to duplicate the evolution of combinatorial logic implemented in semiconductor microelectronics. Present configurations of semiconductor logic gates are well suited for electronic computing but may not be the best choice for optically implemented logic. In

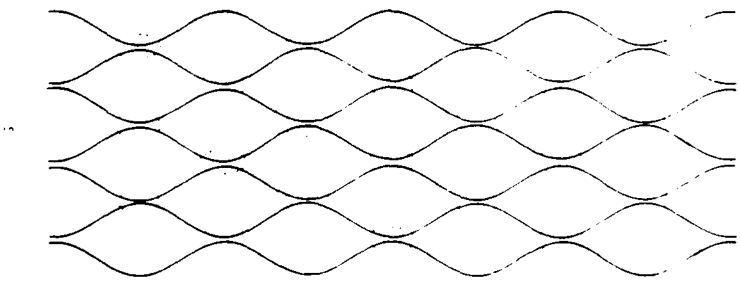


Fig. 7. Integrated optical crossbar. The elliptic regions are the nonlinear coupling switches.

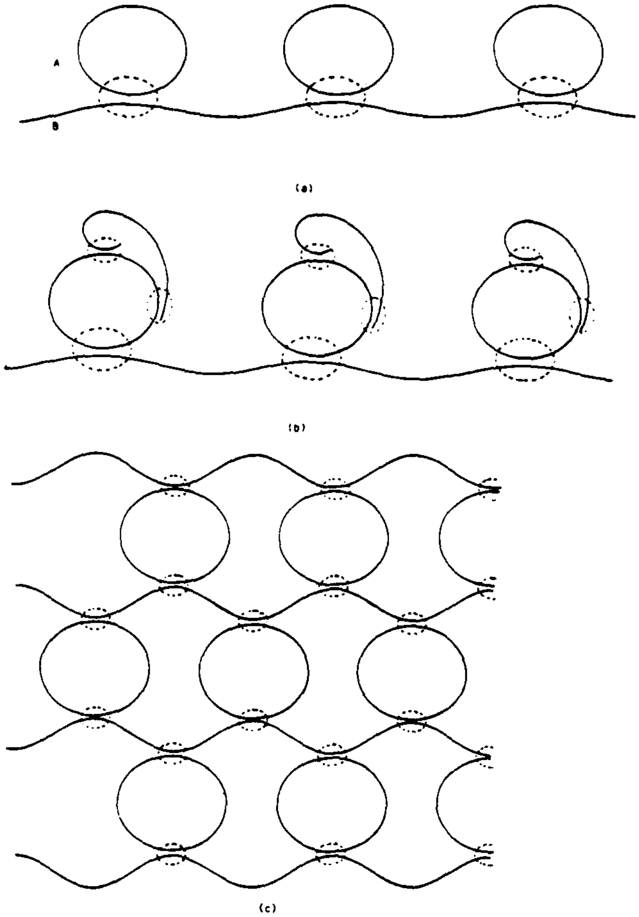


Fig. 8. Tapped delay line:
(a) basic configuration;
(b) reversing modification;
(c) memory array.

this work we have resurrected the concept of reversible logic, illustrated various optical implementations of Fredkin gates, and suggested gate configurations capable of combinatorial logic. These configurations combine the communications advantage of optics with noncapacitive multiline addressing of individual gates and suggest their evaluation as a basic building block for optical computers. The various implementations illustrated here are intended to illustrate the potential of this approach; future work will elaborate on specific higher-order logical functions.

Joseph Shamir is on leave from the Department of Electrical Engineering, Technion—Israel Institute of Technology.

References

1. R. Landauer, "Irreversibility and Heat Generation in the Computing Process," IBM J. Res. Dev. 5, 183 (1961).
2. E. Fredkin and T. Toffoli, "Conservative Logic," Int. J. Theoret. Phys. 21, 219 (1982).
3. See, for example, R. A. Fisher, Ed., *Optical Phase Conjugation* (Academic, New York, 1983).
4. C. S. Tsai, B. Kim, and F. R. El-Akkari, "Optical Channel Waveguide Switch and Coupler Using Total Internal Reflection," IEEE J. Quantum Electron. QE-14, 513 (1978).
5. C. L. Chang and C. S. Tsai, "GHz Bandwidth Optical Channel Waveguide TIR Switches and 4 × 4 Switching Networks," in *Technical Digest*, Topical Meeting on Integrated and Guided-Wave Optics (Optical Society of America, Washington, DC, 1982), paper ThD2.
6. R. A. Forber and E. Marom, "Optimization of Symmetric Zero-Gap Dielectric Couplers for Large Switch-Array Applications," in *Technical Digest, Conference on Lasers and Electro-Optics* (Optical Society of America, Washington, DC, 1985), paper FL2.
7. A. Lattes, H. A. Haus, F. T. Leonberger, and E. P. Ippen, "Title," IEEE J. Quantum Electron. QE-19, 1718 (1983).
8. J. W. Goodman, F. I. Leonberger, S. Y. Kung, and R. A. Athale "Optical Interconnections for VLSI," Proc. IEEE 72, 850 (1984).
9. J. Haavisto and G. A. Pajer, "Resonance Effects in Low-Loss Ring Waveguides," Opt. Lett. 5, 510 (1980).
10. A. Mahapatra and W. C. Robinson, "Integrated-Optic Ring Resonators Made by Proton Exchange in Lithium Niobate," Appl. Opt. 24, 2285 (1985).

Optical Fredkin Gate

Joseph Shamir and H. John Caulfield
Center for Applied Optics, University of Alabama in Huntsville
Huntsville, Alabama 35899

William Miceli
Naval Ocean Systems Center, San Diego, California 92152

Robert J. Seymour
GTE Laboratories Inc., 40 Sylvan Road, Waltham, Mass. 02254

Abstract

Much work is being done toward the optical implementation of traditional electronic processing and computing methods. Many of the proposed methods may not be the optimal way to utilize the benefits of optical techniques. We introduce here a new optical gate - the Fredkin gate - that is in principle minimally dissipative and its response time in some implementations may be limited only by the duration of optical pulses (i.e. in the subpicosecond range). To indicate the viability of this novel approach, a number of optical implementations of Fredkin gates with some interesting applications are proposed.

Introduction

One of the limitations imposed on increasing computation power, be it electronic or optic, stems from the large amount of energy that needs to be dissipated during computer operation[1]. Part of this energy is due to the intrinsic nature of the traditional logic elements. This fact becomes evident if we recall that a conventional logic gate has more input lines than output lines. Thus some of the information coming into the gate is lost and cannot be retrieved. The irreversible nature of the gate makes it dissipative not only in information but also in energy. In an effort to overcome these limitations, Fredkin proposed a new kind of logic gate which has the same number of output lines as it has input lines. Fredkin gates are capable of performing conventional logic operations while preserving all the original information. In contrast to the conventional logic gates the Fredkin gates may, in principle, be run backwards to regenerate the original input signals.

The purpose of this work is to introduce the optical Fredkin gate which may become one of the basic building blocks of an optical computer. An overview of the main aspects of the Fredkin gate is given in the next section, followed by a variety of proposed optical implementations. A number of use applications are discussed in a final section.

Background on the Fredkin Gate

The basic Fredkin gate is defined as a black box having three binary inputs and three binary outputs (Figure 1). The C-input - the control line, determines the operation of the gate on the other two inputs according the following rules:

$$\begin{aligned} &\text{IF} \quad C = 0: \quad A' = A; \; B' = B; \\ &\text{IF} \quad C = 1: \quad A' = B; \; B' = A; \end{aligned} \qquad (1)$$

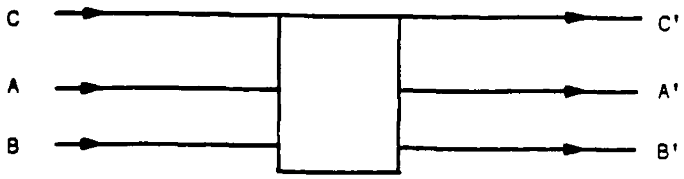


Figure 1.

It is quite evident that this gate is reversible, i.e. it may be run backward to return to the original inputs and therefore it is in principle non-dissipative. (The original definitions used in Ref. 2 are the inverse of Eq. (1); however we find this definition more intuitive and more suitable for optical implementation.)

Using the three inputs and the three outputs of the Fredkin gate one may implement the traditional logic gates that usually have two input lines and one output line. To make the comparison easier, in the examples of Fig. 2 we leave the lines corresponding to the conventional gates straight while the other lines are shown bent. In Figure 2a an AND gate is implemented keeping the A input at the 0 level obtaining the required output on the A; line. Unlike conventional gates, we obtain two additional outputs that we may utilize or ignore. In a similar way, one possible implementation of an OR gate is shown in Figure 2b. It can be easily shown that any other function, such as NOT, FAN-OUT, FAN-IN and FLIP-FLOPs are also easily implemented. In the next section we discuss a number of ways to implement the Fredkin gate by optical and electro-optical means.

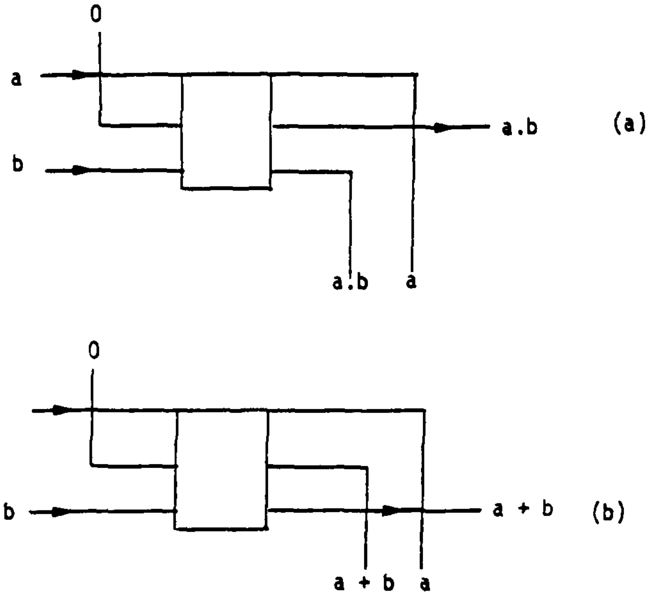


Figure 2.

Optical Implementations of the Fredkin Gate

For applications in logic networks one is usually interested in logic gates containing nonlinear, bistable elements. The basic configuration of a Fredkin gate, however, is not restricted to digital signals and, in principle, one may use these gates for processing analog signals as well. In the examples that follow the nature of the control signal will determine the actual response of the gate.

A polarization switching gate is shown in Figure 3. The A and B lines correspond to two orthogonal polarizations of a light beam (or a waveguide channel of an integrated optical system) traversing an electro-optic modulator that rotates both polarizations by 90° when activated. The activation is induced by a direct electronic pulse or, as shown in the figure, by an optical signal transduced to an electrical signal using a photodetector (Photoconductor or photodiode-amplifier combination). Polarizing beam-splitters

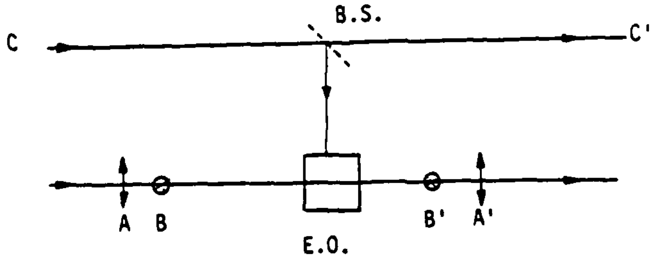


Figure 3.

may be applied whenever a spatial separation is required between the A and B lines. The main advantage of this gate is its relative simplicity while its disadvantage is the different nature of the C-line that also changes level during transition through a gate (i.e. there is a lower light intensity in C' than in C. This effect may, however, be corrected by incorporating an amplifying medium on the line).

In Figure 4 we show a schematic diagram of the acousto-optic gate: The two input lines are laser beams incident on an acousto-optic deflector (either bulk or integrated SAW) at the Bragg angle. If there is no acoustic signal (C = 0), the two beams continue unaffected (A' and B') while if C is present each beam is deflected into the other channel. This is also a simple gate but, here too, one has a C line which is basically basically different in nature than the other two lines. Nevertheless this kind of gate can be easily cascaded and integrated. For example, a single acoustic pulse may activate many gates as it travels along the system.

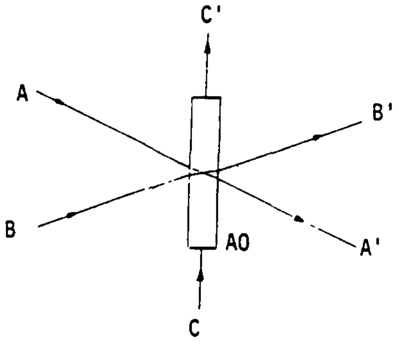


Figure 4.

The Photorefractive gate, based on four-wave-mixing^3 is an all optical gate with one of its tentative implementations illustrated in Figure 5. In this case the C-line constitutes the two pump beams. The inputs A and are transmitted if C is absent and phase-conjugated when the pump is present resulting in a switching between the outputs.

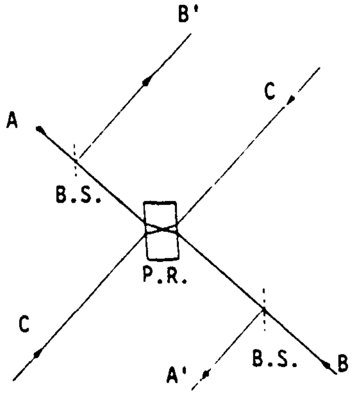


Figure 5.

In optical communication and integrated optical systems a modulated waveguide or fiber coupler may serve as a Fredkin gate. Two general classes of this kind of gates may be implemented. The out-of-plane control, shown schematically in Figure 6a, and the inplane control with one possibility depicted in Figure. 6b. A number of workers have already implemented the electronically addressed coupler [4,5] that may serve as a Fredkin gate with an electronic C-input. To symmetrize the system one may use photodetection combined with the electro-optic coupler to facilitate optical control. A more advanced technology would be the use of photorefractive material for direct optical control of the coupling constant. The example in (b) is a waveguide coupler incorporating highly anisotropic guides containing nonlinear material. The two coupling waves (A and B) are introduced with the same polarization so that they can couple while the control signal, C, is orthogonally polarized in such a way that its power is used to activate the coupling between the A and B channels but it does not couple itself into the other guide[6].

Proposed Devices Incorporating Fredkin Gates

We demonstrate the applicability of these new gates by proposing, in addition to the conventional logic gates, two very useful devices that incorporate arrays of the waveguide gates shown in Figure 6.

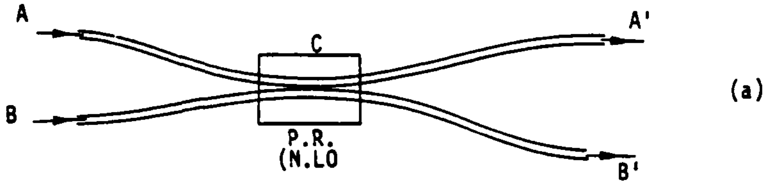


Figure 6

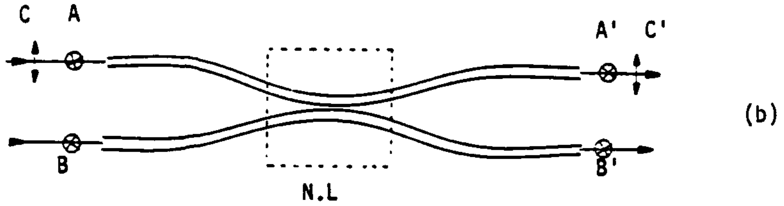


Figure 6.

The optical crossbar. The gate array of Figure 7 may be constructed of gates of the type depicted in Figure 6a or the type of 6b. In the first case each gate may be accessed randomly from above by an electric field or by light, depending on the specific device used[8] As we are dealing with optical computing we might prefer activation by light such as a holographic coupler[8] or fiber coupler. With proper addressing each input line. This system may prove to be an extremely fast and efficient crossbar or optical switchboard. The in-place addressing of Figure 6b is applicable if one desires to activate a whole column together. At first sight it appears that this kind of addressing is not suitable for random access; however with very fast pulses this also becomes feasible.

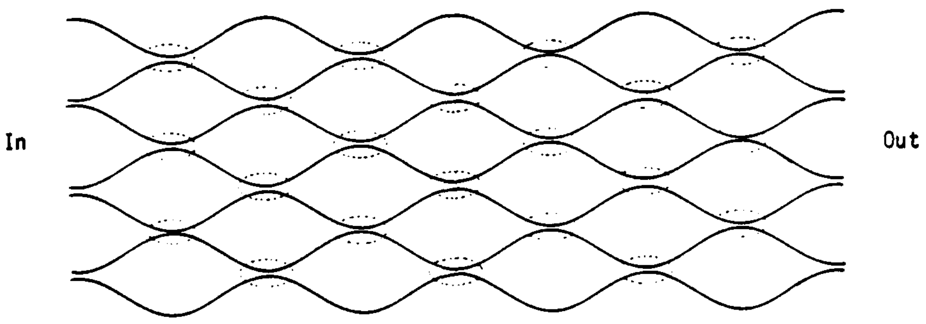


Figure 7.

The tapped delay line. The basic configuration of Figure 8a is a tapped delay line. A fiber ring may be utilized for long delays while for very short delays one may use waveguide rings the feasibility of which has also been demonstrated[9,10]. Here too, the addressing may be of the first type (Figure 6a or of the second (Figure 6b). Such a system may be used to delay all the energy in a pulse or just part of it to produce a pulse train from a single initial pulse. A slight modification of the system as illustrated in Figure 8b may be used to reverse the direction of signal flow resulting in a true reversible Fredkin gate. In the future, an optical memory block may resemble the array depicted in Figure 8c. This seems to be a short term memory, but with the integration of amplifying medium it may serve also as a long-term memory.

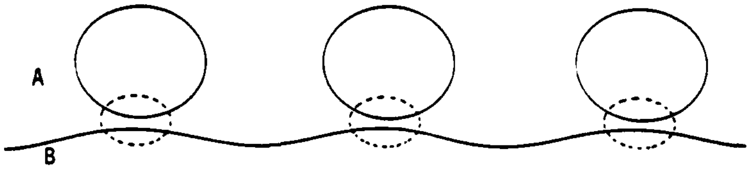


Figure 8(a)

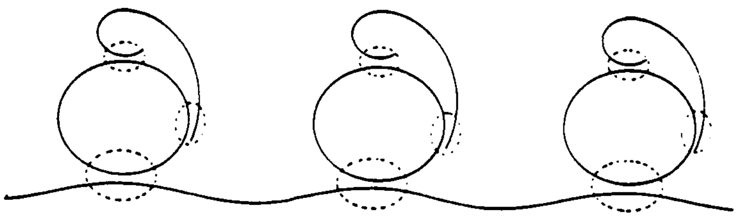


Figure 8 (b)

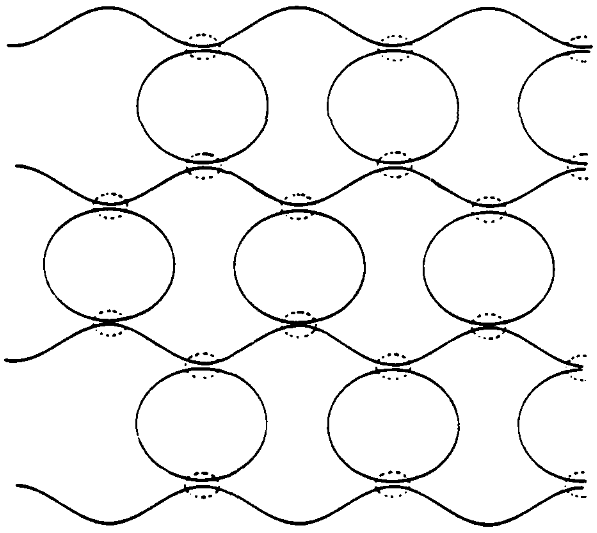


Figure 8(c)

Discussion

Conventional approaches to optical computing followed the lines put forward by workers with electronic systems. Traditional logic gates are well suited for electronic computing but may not be the best choice for optical Processors. In this work we indicated that one should also consider different implementations for optical computing systems with one very promising possibility being the Fredkin gate. These gates have many simple optical implementations and may prove to be very fast and energy efficient. The various implementations and applications given here are just samples to indicate the diverse possibilities available.

References

1. R. Landauer, "Irreversibility and heat generation in the computing process" IBM J. Res. Vol. 5, pp. 183-191, (1961),

2. E. Fredkin and T. Toffoli, "Conservative logic" Int. J. Theor. Phys., Vol. 21, pp. 219-253, (1982).

3. See for example, "Optical phase conjugation"R. A. Fisher ed. Academic Press, N. Y. 1983.

4. C. S. Tsai, B. Kim and F. R. El-Akkari, "Optical channel waveguide switch and coupler using total internal reflection" IEEE J. QE-14, pp. 513-517, (1978).

5. C. L. Chang and C. S. Tsai, "GHz bandwidth optical channel waveguide TIR switches and 4 x 4 switching networks" Topical Meeting on Integrated and Guided-Wave Optics Jan. 6-8, 1982, Pacific Grove, Cal.

6. R. A. Forber and E. Marom. "Optimization of symmetric zero-gap dielectric couplers for large switching-array applications" CLEO, April 1985, Baltimore.

7. A. Lattes, H. A. Haus, F. T. Leonberger, and E. P. Ippen, IEEE J. QE-19, 1718- (1983).

8. J. W. Goodman, F. I. Leonberger, S. Y. Kung and R. A. Athale "Optical interconnections for VLSI" Proc. IEEE Vol. 72, pp. 850-866, (1984)

9. J. Haavisto and G. A. Pajer, "Resonance effects in law-loss ring waveguides" Opt. Lett. Vol. 5, pp 510-512, (1980).

10. A. Mahapatra, and W. C. Robinson, "Integrated-optic ring resonators made by proton exchange in lithium niobate" Appl. Opt. Vol. 22, pp. 2285-2286, (1985).

High-efficiency rapidly programmable optical interconnections

Joseph Shamir and H. John Caulfield

An array of optical Fredkin gates implemented by optically controlled waveguide couplers is showr constitute a very efficient and versatile optical interconnection network with parallel addressing capabilit The characteristics of the array are analyzed using linear algebra to indicate interconnect programr procedures. In terms of SNR this network is estimated to be comparable with previously proposed archi tures. However, from many other aspects (light transmission efficiency, number of switching eleme speed, and fault tolerance) it has significant advantages.

I. Introduction

Optical interconnects were initially investigated for application in integrated electronic processors.[1,5] The demand for highly efficient and fast optical interconnects or programmable crossbars is now increasing with the extensive progress made in the applications of optical fiber communication networks and the expected developments in optical computing. In a recent work[6] the benefits of the optical Fredkin gate were discussed, and several optical implementations were proposed. It was also pointed out there that an array of these gates may function as an optically or electronically addressed optical interconnection network. The array of switching elements building up this network may be addressed in parallel leading to a very fast, light-efficient, and fully programmable device. In principle, the operating speed of the network will be limited by the addressing time, and that may be very short if a page-oriented holographic memory[7,8] is employed. In such a memory bank each useful switching pattern is stored as a hologram that may be addressed by a deflected laser beam.[9,10] Nanosecond addressing time may be possible with an array of 1024 × 1024 holograms.

In the present work we analyze the operation of a general Fredkin gate array interconnection network. Although any optical implementation of the Fredkin gate may be assembled as a useful array, reference h will be made to the most promising one, the optic addressed waveguide coupler array.

Most of the presently demonstrated switcha waveguide couplers employ the electrooptic effect voltage control,[11-14] but direct light-addressa switches are already emerging.[15,16] For our purp we are interested in direct light activated coupl However, a photodetector array connected to an el trooptic switching array will be also quite efficient w its speed limited only by detector delays.[17]

In the next section we describe the architecture the Fredkin gate network with a physical approach the addressing algorithm. The ideal gate array is scribed in Sec. III by a linear algebraic approach e ploying a unitary matrix group. The physical limi tions of a real system is discussed in Sec. IV taking i account losses and crosstalk to evaluate an expec SNR for an actual device. The implementation optical crossbars is addressed in Sec. V with a gene discussion following in Sec. VI.

II. Fredkin Gate Interconnection Network

The basic Fredkin gate is defined as a black l having three binary inputs and three binary outp (Fig. 1). The C-input, control line, determines ope tion of the gate on the other two inputs according to following rules:

$$C = C',$$
$$\text{if } C = 0\text{:}\quad A' = A;\quad B' = B;$$
$$\text{if } C = 1\text{:}\quad A' = B;\quad B' = A.$$

It is evident that this gate is reversible; i.e., it may run backward to return to the original inputs, a therefore, it can be nondissipative, at least in princi Of the various optical implementations proposed Ref. 6 we are interested here in the waveguide cou

The authors are with Center for Applied Optics, University of Alabama in Huntsville, Huntsville, Alabama 35899.

Received 11 October 1986.
0003-6935/87/061032-06$02.00/0.

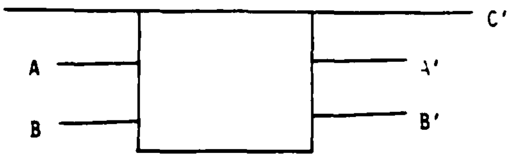
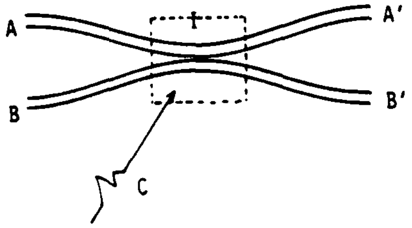


Fig. 1. Fredkin gate.



. Waveguide coupler implementation of the Fredkin gate. *I* .he interaction region where coupling is switched ON or OFF.

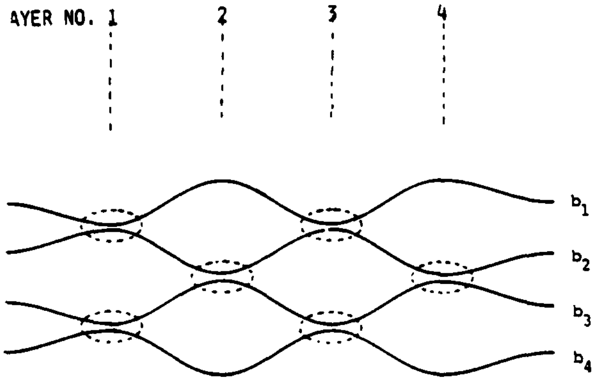


Fig. 3. Four-channel array with four switching layers.

shown in Fig. 2, although any other optical Fredkin gate is applicable. The two inputs, *A* and *B*, are switched when the interaction region *I* is activated by the control signal *C*. The most efficient construction would involve a photorefractive interaction region directly activated by light. However, the electrooptic effect may also be used employing an amplified signal from a photodetector receiving the *C*-input.

The waveguide coupler Fredkin gate of Fig. 2 is our basic building block for constructing a general interconnection network. Figure 3 represents the 4-input and 4-output network. Proceeding from left to right we encounter four layers of interaction regions (numbered 1–4) with each such region activated by an incident control signal. Checking all possible switching combinations one can show that with this arrangement any input signal a_i ($i = 1,2,3,4$) may be coupled into any output port b_i. In other words, all twenty-four permutations are possible with four layers of switches, six switches all together. It is interesting to note that there are forty possible switching states. Thus some of them are redundant with respect to the output configuration. As will be indicated, this redundancy is very useful for fault tolerant operation.

Using induction one may generalize the configuration assuming that for $n = 2N$ channels one needs n interaction layers. If we add two more input channels, a_{n+1} and a_{n+2}, as in Fig. 4, we need two more couplers (the dotted ones in the figure) to switch either of the two new signals into the old array. To make all permutations possible the additional layers should be filled out completely as will be indicated in the mathematical description of the next section. We see that our $n \times n$ network needs n layers with alternating $n/2$ and $n/2 - 1$ switches each. Thus the complete array needs $n(n - 1)/2$ switches to establish all possible interconnections, that is, less than half of the n^2 elements required by most conventional networks.

This whole switching array may be considered as a generalized n-dimensional Fredkin gate: If all control inputs are in the 0 state (all switching elements are

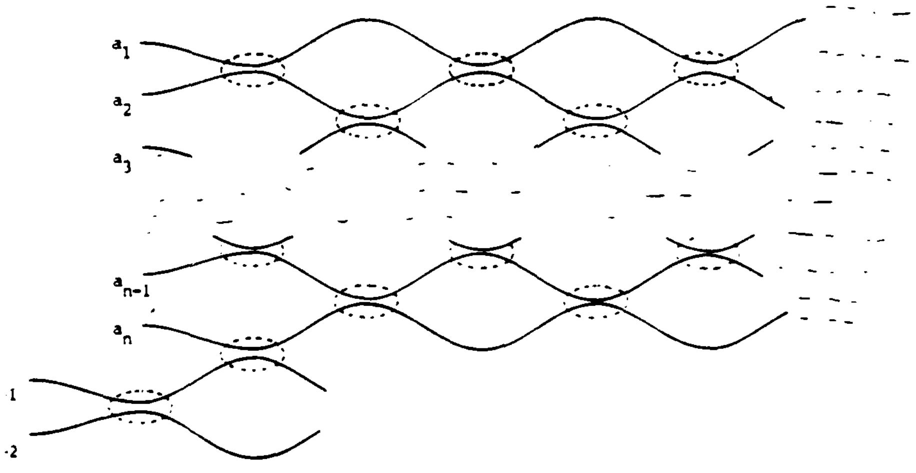


Fig. 4. *n*-channel array with the addition of two more.

OFF) we obtain $b_i = a_i$ for all i, while a complete inversion, i.e., $b_n = a_1$, $b_{n-1} = a_2$, etc., is obtained with all control signals in the 1 state (all switches are ON).

III. Mathematical Analysis

For a mathematical analysis we return to our basic element, the waveguide coupler Fredkin gate of Fig. 2, and represent the input and output channels by vectors **a** and **b**, respectively. The transformation of the vector **a** into the vector **b** may be implemented by a 2 × 2 unitary matrix F(C), where the parameter C may assume the two control values 0 and 1:

$$\mathrm{F}(0) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad \mathrm{F}(1) = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \tag{2}$$

The two possible transformations attainable with this device may thus be written in the matrix form:

$$\mathbf{b} = \mathrm{F}(C)\mathbf{a}. \tag{3}$$

This matrix formalism is easily extendable to a general n-channel device: We observe that each interaction region in a gate layer (see Fig. 3) involves only switching between adjacent channels. Thus, if we describe the input to this layer by the n-element vector **a**, it will be transformed by a block-diagonal unitary matrix

$$\mathrm{P} = \begin{bmatrix} \boxed{\mathrm{F}} & 0\ 0 & \dots & & \\ & 0\ 0 & \dots & & \\ 0\ 0 & \mathrm{F} & 0\ 0 & \dots & \\ 00 & & 0\ 0 & \dots & \\ & & \cdot & & \\ & & \cdot & & \\ & & \cdot & \boxed{} & 0\ 0 \\ & & & 0\ 0 & \boxed{\mathrm{F}} \\ & & & 0\ 0 & \end{bmatrix}; \quad \mathrm{Q} = \begin{bmatrix} 1 & 0\ 0\ 0 & \dots & & \\ 0 & \boxed{\mathrm{F}} & 0\ 0 & \dots & \\ 0 & & 0\ 0 & \dots & \\ 0\ 0\ 0 & & \boxed{\mathrm{F}} & 0\ 0 & \dots \\ & & \cdot & & \\ & & \cdot & & \\ & & \cdot & & \\ & & & \boxed{\mathrm{F}} & 0 \\ & & & \dots 0\ 0 & 1 \end{bmatrix}, \tag{4}$$

where P corresponds to an odd numbered layer, while Q is a matrix corresponding to an even numbered layer. Thus one may describe the complete system transformation by a product of these matrices. The first matrix to operate on the input vector will be P_1, the second matrix will be Q_2, and the final one will be Q_n, and we may write down the complete transformation to be the relation

$$\mathbf{b} = \mathrm{T}\mathbf{a}, \tag{5}$$

where the transfer matrix is

$$\mathrm{T} = \mathrm{Q}_n\mathrm{P}_{n-1}\mathrm{Q}_{n-2}\dots\mathrm{Q}_2\mathrm{P}_1,$$

where the subscripts denote layer numbers.

Equation (6) is a simple mathematical relation th describes any possible switching state of the comple system. To solve the inverse problem, i.e., determi the switching state for a given interconnection, tl requirement is not much more complicated. Since tl $n \times n$ matrix T must have a single element with valu in each row and each column with the rest of tl elements having the value 0, it is a simple matter write down this matrix for any interconnection r quired. The next step is a decomposition of this m trix into n matrices P and Q. This can be done easil since these matrices are the inverse of themselve Thus one can take the T matrix and start multiplyi it by the P and Q matrices until it reduces to the un matrix. If we multiply Eq. (6) from the left by Q_n th matrix is eliminated from the right-hand side of tl equation. The procedure may now go on with tl matrix P_{n-1} etc. until the unit matrix is obtained:

$$\mathrm{P}_1\mathrm{Q}_2\dots\mathrm{P}_{n-1}\mathrm{Q}_n\mathrm{T} = \mathrm{I}, \qquad (6$$

where I is the unit matrix. Physically, the two equ tions (6) demonstrate the reciprocity property of nondissipative optical system.

We clarify the procedure using the six-channel arr illustrated in Fig. 5 with an arbitrarily chosen interco nection pattern indicated on the right-hand side. T write down the transfer matrix we observe that a_3 mu be transferred to the first channel. Thus the first ro should have its unit element in the third place. Sim larly, the second row will have its unit element in tl second column and so on until we construct the who matrix:

$$\mathrm{T} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}. \qquad ($$

Our objective now is to find consecutive P and Q matr ces that will translate all the unit elements into th matrix diagonal. To be most efficient in this proce dure we observe that the units in rows 3 and 4 are at th largest distance from the diagonal, and they can bot

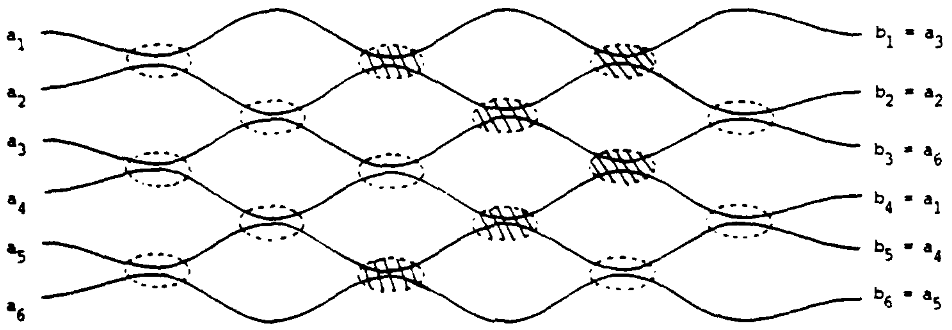


Fig. 5. Six-channel array with selected interconnections.

: brought closer by interchanging them. This goal ay be attained by the P matrix (a Q matrix will do no ›od at this stage):

$$P_5 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

here we also switched between the first two channels nce the unit of the first row is also far from the agonal. Proceeding in this manner, we have

$${}_4 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}; \qquad P_3 = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

hese three matrices complete the task. Thus, for this ıse, only three layers are required to perform the ›eration. This again can be deduced by observing ıat the transfer matrix has its unit elements at a istance from the diagonal not exceeding three posiions. The hatched interaction layers in Fig. 5 desigate the ON elements. This specific example demon:rated also the property of redundancy that may lead › fault tolerance when production limitations are condered.

All the matrices involved until now are unitary ma·ices as we are dealing with ideal nondissipative sys›ms. In the next section we modify the formalism to ıclude losses and leaky switching elements as encoun›red in practice.

/. Real Networks

A real physical network cannot be described by the bove unitary matrices. To take into account losses nd crosstalk in the nonideal switching elements, the asic switching matrix of Eq. (2) should be modified. 'he two states of a real Fredkin gate may thus be epresented by the two modified matrices:

$$F(0) = \begin{bmatrix} 1-\alpha & \beta \\ \beta & 1-\alpha \end{bmatrix} \qquad F(1) = \begin{bmatrix} \gamma & 1-\delta \\ 1-\delta & \gamma \end{bmatrix}, \tag{8}$$

/here α is the loss from the unswitched channel inluding actual loss and leakage β into the second chanıel, and δ is the uncoupled fraction into the switched hannel with γ the fraction of the signal that leaks hrough undeflected. For simplicity a complete symnetry is assumed between the two coupled channels. 'or a working system one naturally must require that $\iota, \beta, \gamma, \delta, \ll 1$. Integrating this gate into an interconıection array returns us to the block-diagonal matrices ›f Eq. (4), but now they are not unitary as they include he lossy matrices [Eq. (8)] instead of the ideal ones of Ξq. (2).

To investigate the effects of the deteriorating pa·ameters we return to the four-channel system of Fig. 4 ınd construct the transformation matrix for one of the most difficult transformations, i.e., a complete inversion with input vector,

$$\mathbf{a}^+ = (1,1,1,0).$$

For this transformation all switching elements are in the ON state. Thus we have to substitute F(1) for all the diagonal blocks in four matrices of the form of Eq. (4). Performing the matrix multiplications and operating on the above input vector yield the output vector

$$\mathbf{b} = \begin{bmatrix} \gamma[\gamma + (1-\delta)^2] + \gamma(1-\delta)[2\gamma + (1-\delta)^2 + (1-\delta)(1+2\gamma)] \\ \gamma^4 + \gamma(1-\delta)(1+\gamma)(2-\delta) + 2\gamma^3(1-\delta) + (1-\delta)^3 \\ \gamma^4 + \gamma(1-\delta)^2(2+\gamma) + (1-\delta)[2\gamma^3 + (1-\delta)^2] \\ (1-\delta)^3 + \gamma(1-\delta)^2(2\gamma+1) + \gamma(1-\delta)[(1-\delta)^2 + \gamma^2 + \gamma] \end{bmatrix}. \tag{9}$$

The ideal transformation would give $b_1 = 0$ with the other three elements 1. Thus we may define a SNR by the relation b_4/b_1 giving

$$\mathrm{SNR} = \frac{(1-\delta)^3 + \gamma(1-\delta)^2(2\gamma+1) + \gamma(1-\delta)[(1-\delta)^2 + \gamma^2 + \gamma]}{\gamma[\gamma + (1-\delta)^2] + \gamma(1-\delta)[2\gamma + (1-\delta)^2 + (1-\delta)(1+2\gamma)]}. \tag{10}$$

Retaining only first-order terms we obtain

$$\mathrm{SNR} = \frac{1-\delta}{3\gamma}. \tag{11}$$

This result could be anticipated since there are three switching elements where a fraction of the unit signal could leak into the zero channel, while the losses from the unit carrying channels are compensated to first order by leakage from the other large-signal channels. Again, by induction, one may generalize this first-order approximation to n channels leading to an expected SNR for a physical network given by

$$\mathrm{SNR} = \frac{1-\delta}{(n-1)\gamma}. \tag{12}$$

Interpreting some experimental results[12-14] one may assume the attainable values, $1 - \delta = 0.95$ and $\gamma = 0.001$ yielding an SNR (>2) up to 500 channels.

V. Optical Fredkin Gate Crossbar

The major function performed by the optical networks described in this work is that of a cross-connector, i.e., the capability to connect any input channel to any output channel. In previously proposed optical crossbars the light input to each channel is spread over all the output channels, and the required connections are obtained by blocking the unwanted connections. From the point of view of the optical design engineer these are blocking crossbars that, for an n-channel system, are only $1/n$ as light efficient as our nonblocking network, where, in an ideal device, all the incident light is utilized for signal transmission. Also, as pointed out earlier, $n(n-1)/2$ switching elements are adequate to perform all interconnections as opposed to n^2 elements in the previous optical crossbars. However, our interconnection network is not completely equivalent to a crossbar.

From the point of view of the network engineer[18] those previously proposed crossbars are nonblocking

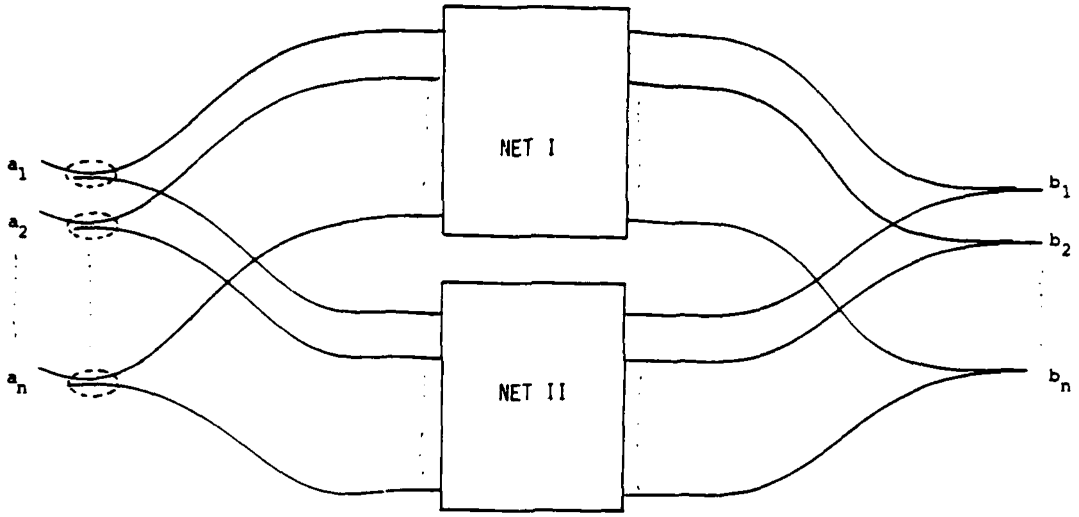


Fig. 6. Nonblocking optical crossbar containing two parallel networks.

in the strict sense in that any idle pair of terminals may be connected without disturbing already established connections. In this sense our interconnector is not a crossbar because one may have to reprogram the whole array to change even a single connection. There is at least one possible solution for the problem that employs two identical networks as shown in Fig. 6. The n controlled directional couplers on the left-hand side are used to switch the whole input pattern between the two networks. If, for example, a new connection is required while information flow is in progress through network I, network II may be programmed to support the complete new connection pattern, and then the inputs may be switched over to network II. In the next occasion the inputs will be switched back to network I. This will be a nonblocking crossbar from the point of view of the optical design engineer as well as from the point of view of the network engineer. Switching between the two networks will not disturb information flow, since during the short transition time both networks will transmit the signals (in complementary amounts of power) that will be combined by the constant directional couplers of the proper output channels on the right-hand-side of the system. It is interesting to point out that the achievement of a strictly nonblocking system was at the expense of additional switching elements returning to the total of n^2.

VI. Discussion

The optical Fredkin gate was shown to be an excellent Building block for construction of a programmable optical interconnection array. Such an array can perform all interconnection requirements, such as the function of a crossbar or perfect shuffle. The overall performance should be significantly better than any other approach proposed until now. Using page oriented holographic memories this will be the fastest programmable interconnection network constructed, and, in most cases, it needs only half of the activ elements of any other configuration.

Being nonblocking with respect to light manipulation results in additional benefits: All the light energ coupled into the system is being extracted as signal except for the inevitable losses encountered in an physical system. Furthermore, if a defective switcl exists in the array, the signal will in most cases b transmitted without deflection. This characteristic together with the implicit redundancy in the systen may be utilized for fault tolerant operation. For example, assume that there is an anticipated fraction e o faulty switching elements introduced during the manufacturing process of an n-channel array. To avoi the faulty elements it is a simple matter to make a array that has $n(1 + e)$ channels (and switching layers and then ignore the faulty layers during programming The introduction of additional channels can also support the solution of problems such as FAN-OUT an FAN-IN.[19]

Considering the problem of signal deterioration i was shown that the Fredkin gate network should perform comparable with an optically blocking networ that has a constant SNR similar to the worst case SN in the present system.

In conclusion, one may state that the optical Fredki gate array may turn out to become the best solution fo the implementation of optical interconnections. Recalling the fact that these gates can also perform logi operations[6] they should be seriously considered as th basic building blocks for a future digital optical computer.

This work was partially supported by the Office o Naval Research under contract NOOO14-86-K 0591.

Joseph Shamir is on leave from the Department o Electrical Engineering of the Technion—Israel Institute of Technology.

References

1. J. W. Goodman *et al.* "Optical Interconnections for VLSI Systems," Proc. IEEE **72**, 850 (1984).
2. R. K. Kostuk, J. W. Goodman, and L. Hesselink, "Optical Imaging Applied to Microelectronic Chip-to-Chip Interconnections," Appl. Opt. **24**, 2851 (1985).
3. A. A. Sawchuk and B. K. Jenkins, "Dynamic Optical Interconnections for Parallel Processing," Proc. Soc. Photo-Opt. Instrum. Eng. **625**, 143 (1986).
4. A. W. Lohmann, "What Classical Optics can do for the Digital Optical Computer," Appl. Opt. **25**, 1543 (1986).
5. E. Marom and N. Konforti, "Programmable Optical Interconnects," Soc. Photo-Opt. Instrum. Eng. **700**, 296 (1986).
6. J. Shamir, H. J. Caulfield, W. Miceli, and R. J. Seymour, "Optical Computing and the Fredkin Gate, Appl. Opt. **25**, 1604 (1986).
7. T. M. Gaylord, "Digital Data Storage," in *Handbook of Optical Holography*, H. J. Caulfield, Ed. (Academic, New York, 1979), pp. 379–413.
8. M. M. Mirsalehi and T. K. Gaylord, "Truth-Table Look-up Parallel Data Processing using an Optical Content-Addressable Memory," Appl. Opt. **25**, 2277 (1986).
9. U. J. Schmidt, "Present State of the Digital Laser Beam Deflection Technique for Alphanumeric and Graphic Displays," in *Progress in Electro-Optics*, E. Camatini, Ed. (Plenum, New York, 1975), pp. 161–180.
10. E. H. Young and S.-K. Yao, "Design Considerations of Acoustooptic Devices," Proc. IEEE **69**, 54 (1981).
11. L. McCaughan and G. A. Bogert, "4 × 4 Ti:LiNbO3 Integrated Optical Crossbar Switch Array," Appl. Phys. Lett. **47**, 348 (1985).
12. R. Chen and C. S. Tsai, "Thermally Annealed Single-Mode Proton-Exchanged Channel-Waveguide Cutoff Modulator," Opt. Lett. **11**, 546 (1986).
13. S. K. Korotky *et al.*, "Fully Connectorized High-Speed Ti:LiNbO3 Switch/Modulator for Time-Division Multiplexing and Data Encoding," IEEE/OSA J. Lightwave Technol. **3**, 1 (1985).
14. R. A. Becker and W. S. C. Chang, "Electrooptical Switching in Thin Film Waveguides for a Computer Communication Bus," Appl. Opt. **18**, 3296 (1979).
15. D. F. Clark, I. Andonovic, and B. Culshaw, "Perturbation Analysis for the Design of an Optically Controlled Fiber-Optic Directional Coupler," Opt. Lett. **11**, 540 (1986).
16. L. A. Molter-Orr and H. A. Haus, "Multiple Coupled Waveguide Switches Using Alternating $\Delta\beta$ Phase Mismatch," Appl. Opt. **24**, 1260 (1985).
17. B. Clymer and J. W. Goodman, "Optical Clock Distribution to an IC Chip," Opt. Eng. **25**, 1103 (1986).
18. See, for example, C. Clos, "A study of Non-blocking Switching Networks," Bell Syst. Tech. J. **32**, 406 (1953).
19. J. W. Goodman, "Fan-in and Fan-out with Optical Interconnections," Opt. Acta **32**, 1489 (1985).

■ The **AIP Center for History of Physics** continues its grants-in-aid of up to $1,000 to reimburse direct expenses for research in the history of 19th and 20th century physics, geophysics, astronomy and allied sciences, and their social interactions. Applications should be sent to Spencer Weart, AIP, 335 East 45th St., New York, NY 10017.

Three-dimensional optical interconnection gate array

Joseph Shamir

A recently proposed planar Fredkin gate array for optical interconnections is extended here into a 3-D array that can be implemented using ferroelectric liquid crystal spatial light modulators. Operating as polarization gates these modulators are efficient and can be incorporated into high performance interconnection networks. Some advantages of the new architecture are discussed and performance characteristics are estimated.

I. Introduction

One of the most promising uses of optics in computing and communications is the implementation of complicated interconnections. For this application, planar architectures of Fredkin gate arrays constructed of optical waveguide couplers were recently investigated.[1] This work demonstrated that these arrays are efficient with respect to the utilization of light power and they are rapidly programmable. In addition to their use in optical interconnection networks these arrays can also be employed in various processing operations such as residue arithmetics,[2] logic gate arrays, and variable delay lines. The planar configuration is attractive for applications in conjunction with integrated optical and electronic devices; however the advantages of the 2-D parallelism possible with optical systems were not fully exploited. In the present work we explore the performance of 3-D architectures and indicate their implementation using polarization Fredkin gate arrays.

II. Planar Interconnection Network

We start with a short review of the planar interconnection array of n channels that was investigated in Ref. 1. One possible implementation of such an array employs controllable waveguide couplers as represented schematically in Fig. 1 for a seven-channel waveguide array. In this array there are seven channels with respective input signals, a_i $(i = 1,\ldots 7)$, seven outputs, b_i, and seven layers of couplers (switches) that are either OFF or ON. When a switch is in the OFF state the signal in each channel is transmitted through the coupling region and remains in its original channel. With the switch in the ON state the signals are interchanged between adjacent channels. It was shown in Ref. 1 that, for such a configuration containing n channels $(i = 1,\ldots n)$, one may obtain b_i with all the possible permutations of a_i using n switching layers with a total number of $n(n-1)/2$ switches. In a practical situation where the switching elements are not ideal one may assign some average parameter, γ, for the fraction of the signal that leaks through the coupler into the unwanted channel and obtain an approximate value[1] for the signal-to-noise ratio (SNR) at the output:

$$\mathrm{SNR} = \frac{1-\gamma}{(n-1)\gamma} . \tag{1}$$

III. Three-Dimensional Arrays

To improve the performance of the system by exploiting the 2-D capabilities of an optical system one may stack m-planar arrays (such as in Fig. 1), each of n channels, into a 3-D architecture [Fig. 2(a)]. The switching layers are now arranged as matrices over transversal planes but each planar array is independent of the others. Extending the earlier analysis, it is easy to see that for the stack of n-channel arrays one needs n-switching layers to perform all possible horizontal interconnections. To make all vertical connections available too, we augment the configuration by n vertically oriented planar arrays [Fig. 2(b)] of m channels each, containing m-switching layers. Thus a complete interconnection array [a cascade of Figs. 2(a) and (b)] can be implemented using m layers of $n(n-1)/2$ switching elements and n layers with $m(m-1)/2$ switching elements summing up to a total of

$$N = mn(n-1)/2 + nm(m-1)/2 = mn(n+m-2)/2 \tag{2}$$

switching elements. With a square array of n^2 chan-

When this work was done the author was with University of Alabama in Huntsville, Center for Applied Optics, Huntsville, Alabama 35899; he has now returned to Technion—Israel Institute of Technology, Department of Electrical Engineering, Haifa 32000, Israel.

Received 30 January 1987.
0003-6935/87/163455-03$02.00/0.

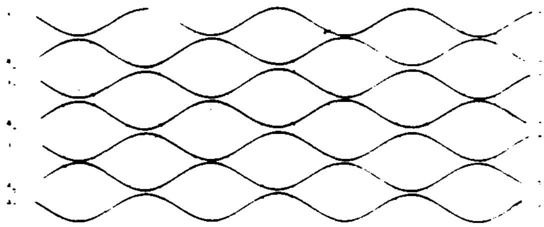


Fig. 1. Seven-channel planar interconnection array.

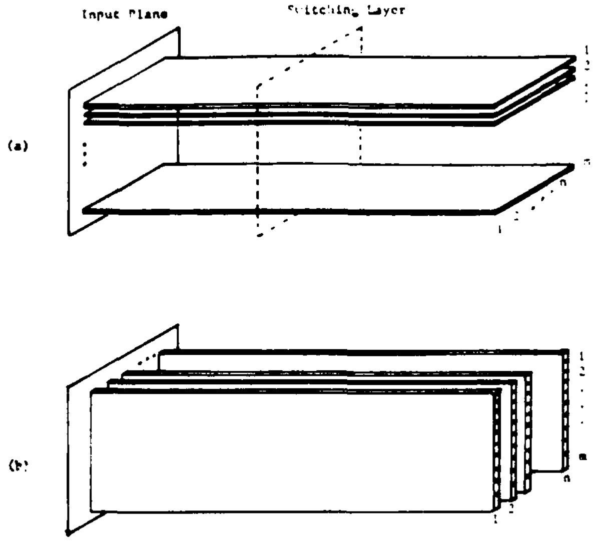


Fig. 2. (a) Stack of m planar arrays of n horizontal channels each. (b) Stack of n planar arrays of m vertical channels each.

nels ($m = n$) one needs $n^2(n - 1)$ switching elements compared with n^4 of a regular planar crossbar. It should be pointed out, however, that FAN-IN and FAN-OUT operations with this simple configuration are possible only at the expense of additional channels as indicated in Ref. 2.

Regarding the SNR, one may repeat the calculations[1] that lead to Eq. (1) or just observe that, to first order, it is inversely proportional to the number of switching layers. Thus in our case we may write, instead of Eq. (1),

$$\mathrm{SNR} = \frac{1 - \gamma}{(n + m - 1)\gamma}, \tag{3}$$

which is an appreciable improvement compared with the planar array where for M ($= n \times m$ in the present case) channels the sum in the denominator would have to be replaced by the product ($n \times m$). Waveguide arrays as described in Ref. 1 are ideal for planar networks; however, for this 3-D architecture different kinds of device may prove more useful.

IV. Polarization Gate Arrays

In Refs. 3 and 4 polarization logic gates were proposed while, independently, in Ref. 5, a similar model was proposed for the implementation of optical Fredkin gates. In these gates the switching operation rotates the polarization of an incident beam by 90°. If employed as a logic gate one polarization is defined as a logic 1 while the orthogonal polarization is defined as a logic 0. In this work we use the Fredkin gate definition: the two orthogonal polarizations represent the two separate input channels to the gate while the switching operation interchanges the two channels. In each of these channels the presence or absence of a signal indicates the logic 1 and 0, respectively.

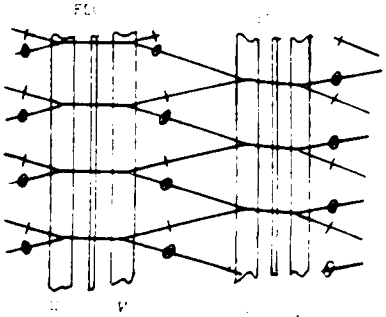


Fig. 3. Section of the ferroelectric interconnection array; FLC, ferroelectric gate array; W, Wollaston prisms. Polarization of the various channels is indicated.

Polarization Fredkin gates may be implemented by various electrooptic or magnetooptic modulators. For the present purpose, the most promising device is the ferroelectric liquid crystal spatial light modulator[6] (FLC) that already exists in the form of large arrays. Each pixel of the FLC can be addressed separately to switch ON or OFF a halfwave retardation, thus performing the requirement of a polarization Fredkin gate.

The top view of a section of the proposed polarization interconnection array is shown schematically in Fig. 3. A suitably designed Wollaston prism is employed to combine two channels into a single gate element (pixel). After transmission through the gate a second, similar Wollaston prism separates the two polarizations (channels) and directs them toward two adjacent gates in the next stage that is shifted transversally by half of the distance between pixels. The layout of each horizontal plane resembles the planar waveguide array of Fig. 1, and each FLC sandwiched between two Wollaston prisms performs the function of a 2-D coupling array as required in Fig. 2(a). Using FLCs in arrays of $n \times m$ we may implement the complete interconnection network with $2n$ stages (each pixel in the FLC represents two signal channels) performing the horizontal interconnection between the $2n$ channels similar to Fig. 2(a). To implement the vertical interconnections required in the architecture of Fig. 2(b) one needs $2m$ additional stages with the Wollaston prisms rotated by 90°.

To estimate the SNR of an interconnection network one may use the reported switching contrast ratio of ~100:1. Deducing from this a signal leakage value of ~0.01 we obtain for a square array of $n \times n$ gates ($4n \times n$ channels) by Eq. (3),

$$\mathrm{SNR} = \frac{0.99}{0.01 \times (4n - 1)}. \tag{4}$$

Thus the SNR with presently available gate arrays will be better than 2 up to $n = 12$, i.e., a total of ~500 channels that can be switched at a rate approaching 1 MHz. Research on this kind of spatial light modulator indicates that the above numbers may be appreciably improved in the future.

V. Conclusions

Exploiting the 2-D addressing capabilities in optical systems, it was shown that efficient programmable interconnection networks can be implemented in a 3-D architecture. Even using existing liquid crystal spatial light modulators that were not designed for the present purpose, high density and low-loss networks are possible. In addition to their application in interconnection networks these arrays may become useful in other fields, such as optical logic gate arrays, arithmetic processors, programmable delay lines, phased arrays, and wideband signal analyzers.

It is a pleasure to thank K. M. Johnson for stimulating discussions about the ferroelectric liquid crystal gate arrays.

This work was partially supported by the Office of Naval Research under contract N00014-86-K-0591.

References

1. J. Shamir and H. J. Caulfield, "High-Efficiency Rapidly Programmable Optical Interconnections," Appl. Opt. **26**, 1032 (1987).
2. M. M. Mirsalehi, J. Shamir, and H. J. Caulfield, "Residue Arithmetic Processing Utilizing Optical Fredkin Gate Arrays," Appl. Opt. 15 Sept. '87, to be published.
3. A. Korpel and A. W. Lohmann, "Polarization and Optical Bistability," Appl. Opt. **25**, 1528 (1986).
4. A. W. Lohmann and J. Weigelt, "Spatial Filtering Logic Based on Polarization," Appl. Opt. **26**, 131 (1987).
5. J. Shamir, H. J. Caulfield, W. J. Micelli, and R. J. Seymour, "Optical Computing and the Fredkin Gates," Appl. Opt. **25**, 1604 (1986).
6. L. A. Pagano-Stauffer, K. M. Johnson, H. J. Masterson, N. A. Clark, and M. A. Handschy, "Optical Logic Gates Using Ferroelectric Liquid Crystals," J. Opt. Soc. Am. A 3(13), P105 (1986).

OSA R.W. WOOD PRIZE
David E. Aspnes

David E. Aspnes of Bell Communications Research is the 1987 winner of the R.W. Wood Prize of the Optical Society of America. Established in 1975 and presented annually, the award recognizes an outstanding discovery, scientific or technological achievement, or invention in the field of optics. Aspnes, an OSA fellow, is being cited for his "pioneering role in the development of the technique of spectroscopic ellipsometry as a probe for materials characterization."

Since 1983, Aspnes has been a district manager at Bell Communications Research in Murray Hill, NJ. He joined the technical staff at AT&T Bell Laboratories in 1967, establishing a research program on the optical properties of solids, surfaces, and interfaces. The experimental and theoretical work led initially to the discovery and refinement of low-field electroreflectance, now the standard approach to high-resolution energy band and compositional spectroscopy of semiconductors and alloys. While Aspnes's work has emphasized optical spectroscopy, his accomplishments include discovery of a new class of focusing mirrors and grazing incidence monochromators for use in vacuum UV and soft x-ray optical systems.

During 1976, he was a senior scientist at the Max-Planck-Institute for Solid-State Research in Stuttgart, F.R.Germany. He received B.S. and M.S. degrees in electrical engineering from the University of Wisconsin, Madison, in 1960 and 1961, respectively. In 1965, he was awarded a doctorate in physics from the University of Illinois, Urbana. Aspnes is also a fellow of the American Physical Society.

The Wood Prize consists of a medal, a scroll, and $1000. Aspnes will receive it at Optics '87 (the OSA annual meeting), to be held this year during October in Rochester, NY.

Reprinted from Applied Optics, Vol. 26, page 3940, September 1 , 1987

Residue arithmetic processing utilizing optical Fredkin gate arrays

Mir M. Mirsalehi, Joseph Shamir, and H. John Caulfield

A cascadable residue arithmetic processor based on optical Fredkin gate arrays and page-oriented holographic memories is introduced. The implementations of residue functions and operations by this processor are described. Analytic expressions are derived for the number of holograms and waveguide channels required for the implementation of residue addition and multiplication. The practical cases of 16-bit addition and multiplication are analyzed as specific examples. It is shown that, using the proposed architecture, these operations can be implemented with state-of-the-art technologies in holography and integrated optics.

I. Introduction

There is a growing interest in the field of digital optical computing.[1] To obtain digital optical processors that greatly surpass the performance of the present computers, the inherent advantages of optics should be utilized. Two major advantages of optics are interconnection and parallelism. Global interconnections can be achieved by classical optical devices, such as prisms and lenses,[2] or by holograms.[3,4] Also, it has been recently shown that an array of optical Fredkin gates constitutes a very efficient and versatile interconnection network.[5,6] Parallel processing can be achieved easily in optics by manipulating the elements of a 2-D array. To take full advantage of the parallelism in optics, digital techniques that are suitable for parallel processing can be utilized. One of these techniques is residue arithmetic, which is based on the residue number system (RNS). The main advantage of the RNS is that its digits are independent of each other; e.g., there is no carry in addition. This allows simultaneous operation on all digits.

The purpose of this paper is to show how an array of optical Fredkin gates can be used to realize residue arithmetic. To provide the required background, residue arithmetic and Fredkin gates are briefly described in Sec. II. The general realization of residue arithmetic with optical Fredkin gates is introduced in Sec. III, while the implementation of residue addition, multiplication, and other operations are described in Secs. IV, V, and VI. Finally, in Sec. VII, the potential characteristics of this processor are summarized.

II. Background on Residue Arithmetic and Fredkin Gates

A. Residue Arithmetic

The foundation of residue arithmetic dates back to the first century A.D., when the Chinese mathematician Sun-Tsu published a verse in which he gave an algorithm for finding a number whose remainders on division by 3, 5, and 7 are known. A general theory of remainders (now known as the Chinese remainder theorem) was established by the German mathematician K. E. Gauss in the nineteenth century. The application of residue arithmetic in computers, however, is relatively recent and was first introduced in 1955 by Svoboda and Valach in Czechoslovakia.[7]

Unlike the commonly used binary and decimal number systems, the residue number system (RNS) is an unweighted system. The base of a residue system consists of n pairwise relatively prime (having no common factor) numbers, $m_1, m_2, \ldots, m_n$, called moduli. Any integer X can then be represented by an n-tuple $(x_1,x_2, \ldots ,x_n)$, where $x_i = |X|_{m_i}$, (read X mod m_i) is the positive remainder that is obtained from the division of X by m_i. This representation is unique for a dynamic range of

$$M = \prod_{i=1}^{n} m_i.$$

An important feature of the RNS is that the fixed-point arithmetic operations can be performed on each digit individually. That is, if $X = (x_1,x_2, \ldots ,x_n)$ and $Y = (y_1,y_2, \ldots ,y_n)$ are two numbers of the same residue system, $Z = X * Y = (z_1,z_2, \ldots ,z_n)$, where $z_i = |(x_i * y_i)|_{m_i}$ for $i = 1, 2, \ldots, n$, and $*$ represents addition, subtraction, or multiplication. Division can be performed, but it is difficult except for the remainder zero case.[8]

As an example, consider the set of four moduli $\{5, 7, 8, 9\}$. These moduli cover a dynamic range of 2520. In this residue system, the decimal numbers $X = 42$ and $Y = 31$ are represented as $X = (2,0,2,6)$ and $Y = (1,3,7,4)$.

The authors are with the University of Alabama in Huntsville, Huntsville, Alabama 35899. Mir Mirsalehi is in the Electrical & Computer Engineering Department; the other authors are in the Center for Applied Optics.

Received 13 December 1986.
0003-6935/87/183940-07/$02.00/0.

The results of performing addition, subtraction, and multiplication on these numbers are $X + Y = (3,3,1,1)$, $X - Y = (1,4,3,2)$ and $X \cdot Y = (2,0,6,6)$, which are the residue representations of the correct answers, i.e., 73, 11, and 1302, respectively.

B. Fredkin Gates

The basic Fredkin gate has three binary inputs and three binary outputs (Fig. 1). The control input C determines the operation of the gate according to the following rules:

$$\begin{aligned} &C' = C, \\ &A' = A \text{ and } B' = B, \quad \text{if } C = 0, \qquad (1)\\ &A' = B \text{ and } B' = A, \quad \text{if } C = 1. \end{aligned}$$

The Fredkin gate is a functionally complete set in Boolean algebra. That is, any binary logic operation, such as AND, OR, and NOT, can be realized by Fredkin gates. The application of optical Fredkin gates as interconnecting systems is of special interest. It has been shown recently that an array of optical Fredkin gates can operate as a very efficient interconnection network with parallel addressing capabilities.[6]

Optical Fredkin gates can be implemented by various techniques.[5] Here we are interested in the implementation by the waveguide couplers shown in Fig. 2. The two inputs, A and B, are switched when the interaction region I is activated by the control signal C. The most efficient construction involves a photorefractive interaction region directly activated by light. However, the electrooptic effect may also be used by employing an amplified signal from a photodetector that receives the C input.

The waveguide coupler of Fig. 2 is the basic building block for constructing a general interconnection network. As an example, a four-input, four-output network is shown in Fig. 3. Checking all possible switching combinations, one can show that with this arrangement any input signal a_i ($i = 1, 2, 3, 4$) may be coupled into any output port b_i. In other words, all twenty-four permutations of the four inputs are possible with four layers of switches and a total of six switches. In general, for $n = 2N$ channels, one needs n interaction layers and $n(n - 1)/2$ switches to establish all $n!$ possible permutations of the inputs.

III. Implementation of Residue Arithmetic by Optical Fredkin Gates

The calculations in residue arithmetic have a cyclic nature. Therefore, they can be implemented by physical properties that are also cyclic in nature. Using the cyclic property of the phase or polarization of light, optical residue-based processors have been developed.[9-12] A major problem with these implementations is that precise control of the phase or polarization of light is usually difficult and requires bulky devices.

A better technique is to use positional coding for data representation.[13-18] The input and output of a residue processor modulo m can have integer values from zero to $m - 1$. Since the modulus is usually a small number, it is practical to have m channels corresponding to these values. An input number is then coded as the presence of light in the channel that corresponds to its value. Any process on the input data is possible by coupling the light from the input channel to the appropriate output channel using an interconnecting system.

As illustrative examples, two interconnections that implement residue functions modulo four are shown in Fig. 4. The system in Fig. 4(a) adds two to an input number in residue arithmetic. Using modulo four, the possible values of the input number are 0, 1, 2, and 3. With the above operation, these values are mapped to 2, 3, 0, and 1, respectively. Figure 4(b) shows a system that provides the third power of a residue number modulo 4. Other residue functions can be realized by similar interconnections.

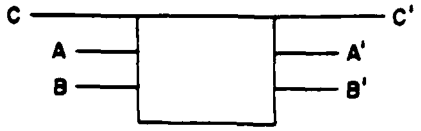


Fig. 1. Fredkin gate.

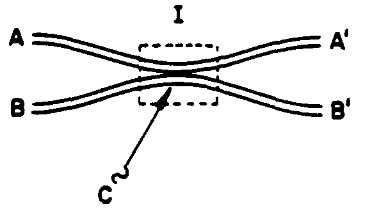


Fig. 2. Waveguide coupler implementation of the Fredkin gate. I is the interaction region where coupling is switched ON or OFF.

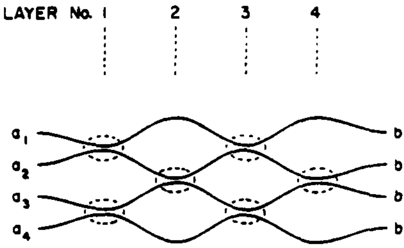


Fig. 3. Fredkin gate array of four channels and four switching layers.

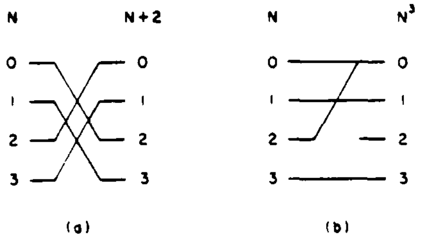


Fig. 4. Example implementations of functions in residue arithmetic by interconnecting systems: (a) addition of 2 to a residue number modulo 4; (b) raising a residue modulo 4 number by power 3. The input is entered from the left, and the output is obtained from the right.

Optical Fredkin gates in conjunction with page-oriented holographic memories can be used to implement the interconnections required for residue arithmetic. Figure 5 shows such a processor that uses modulo 4. Starting from the top, the four channels correspond to integers 0, 1, 2, and 3. Depending on the processes of interest, a number of holograms are recorded at different locations of a holographic material. The input number is coded as the presence of light in one of the input channels on the left, and a laser beam is deflected to a particular hologram corresponding to the required process. The reconstructed beams activate some of the switching elements coupling the light from the input channel to the appropriate output channel.

The above processor can be realized with present technology. Optical waveguide couplers can be fabricated using integrated optics technology.[19] Different holographic materials such as photographic films, dichromated gelatin, thermoplastic materials, or photorefractive crystals can be used for recording.[20] Finally, the deflection of the laser beam can be achieved by an acoustooptic cell.[21] With the progress in the technology of spatial light modulators, they may replace the combination of the acoustooptic deflector and hologram. However, their operation will be relatively slow. In the following two sections, the implementations of residue addition and multiplication with this architecture are analyzed in more detail.

IV. Residue Addition

To implement a residue operation on two numbers, one of the numbers N_1 is used as the input to the system, while the other number N_2 is used for selecting the proper interconnection. To illustrate this point, Fig. 6 shows the four types of interconnection (maps) that are needed for implementing residue addition modulo 4. One of these maps [Fig. 6(a)] is a straightthrough interconnection which can be obtained by default; there is no need to activate any switches. Each of the other three interconnections can be realized by activating some of the switches. Therefore, the whole residue addition modulo 4 operation can be implemented with four channels and only three holograms (Fig. 7). In general, the implementation of residue addition modulo m requires a Fredkin gate array of m channels and m layers, thus $m(m-1)/2$ switches, and recording $m-1$ holograms.

In practical cases, a digital system should have a large dynamic range. This can be achieved by choosing a set of pairwise relatively prime moduli. The optimum set of moduli, in the sense of covering the required dynamic range with minimum number of holograms, consists of numerous small moduli which are either prime or powers of prime numbers. The procedure for selecting such a set of moduli for a required dynamic range can be found in Ref. 22.

As an example, a 16-bit fixed-point operation requires a dynamic range of $2^{16} = 65{,}536$. The optimum set of moduli for this case is {3, 5, 7, 8, 11, 13}, which covers a dynamic range of 120,120. Each modulus m_i is treated individually by devoting m_i channels to it.

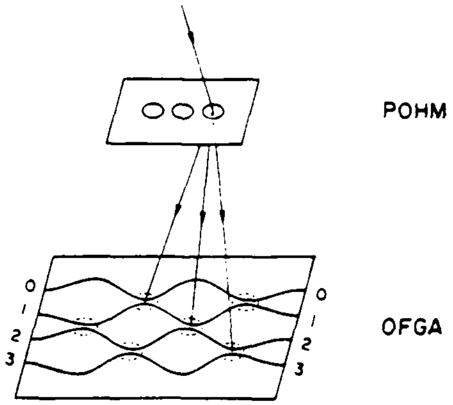


Fig. 5. Schematic diagram of the proposed processor: POHM, page-oriented holographic memory; OFGA, optical Fredkin gate array.

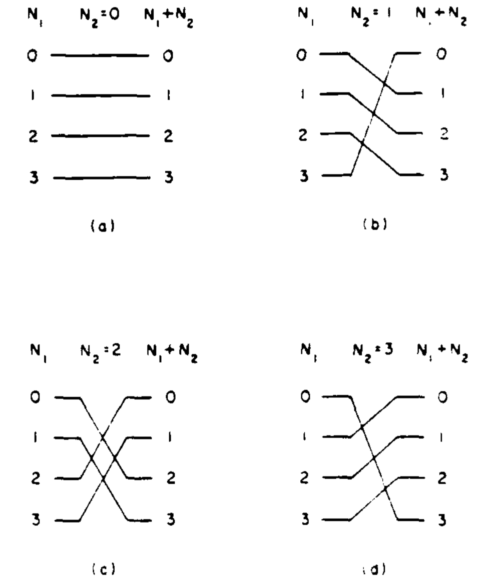


Fig. 6. Interconnections corresponding to residue addition ($N_1 + N_2$) modulo 4. The interconnections (a), (b), (c), and (d) correspond to $N_2 = 0, 1, 2$, and 3, respectively. The input N_1 is entered from the left, and the output $N_1 + N_2$ is obtained from the right.

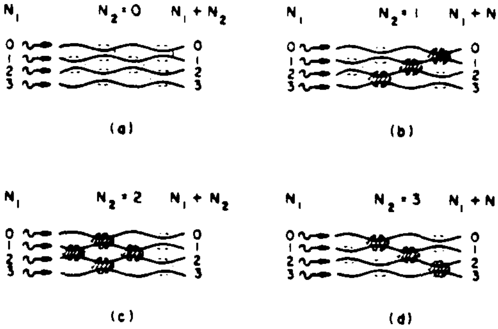


Fig. 7. Required switching states for implementing residue addition ($N_1 + N_2$) modulo 4. The hatched switching elements are ON. The four interconnections realized in (a), (b), (c), and (d) correspond to $N_2 = 0, 1, 2$, and 3, respectively.

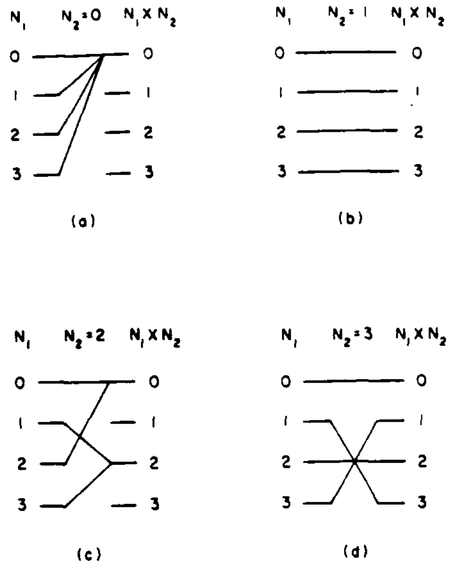


Fig. 8. Interconnections corresponding to residue multiplication ($N_1 \times N_2$) modulo 4. The interconnections (a), (b), (c), and (d) correspond to $N_2 = 0$, 1, 2, and 3, respectively. The input N_1 is entered from the left, and the output $N_1 \times N_2$ is obtained from the right.

Considering residue addition, the number of required holograms corresponding to moduli 3, 5, 7, 8, 11, and 13 are 2, 4, 6, 7, 10, and 12, respectively. The number of switches corresponding to the above moduli are 3, 10, 21, 28, 55, and 78, respectively. Therefore, the 16-bit fixed-point addition can be implemented in residue arithmetic by a page-oriented holographic memory consisting of forty-one patterns and a waveguide gate array consisting of forty-seven channels and 195 switches.

V. Residue Multiplication

The implementation of residue multiplication by Fredkin gates is not as easy as the residue addition case. This is due to the difference that exists between the types of interconnection needed for these operations. Residue addition has the property that each possible value has the same number of occurrences in the output. Also, the mappings corresponding to residue addition are one-to-one (onto). These properties are not valid for residue multiplication.[23] For example, the four interconnections corresponding to residue multiplication modulo four are shown in Fig. 8. It can be seen that the occurrences of the output values are not the same and that two of the mappings [(a) and (c)] are not one-to-one. Using Fredkin gate arrays, any permutation of the input signals can be achieved. However, no two input signals can be coupled into the same output port. Therefore, Fredkin gate arrays are naturally suitable for onto mappings, and some modifications are required to implement a general case as described in the following subsections.

A. Increasing the Number of Holograms

One method for implementing residue multiplication with optical Fredkin gates is to increase the number of holograms. The selection of the appropriate hologram for a specific case then depends on both input numbers. As an example, we discuss residue multiplication modulo 4. The realization of the interconnection for the $N_2 = 0$ case [Fig. 8(a)] requires the recording of three holograms corresponding to $N_1 = 1$, 2, and 3. The case of $N_1 = N_2 = 0$ does not need a hologram, since zero-to-zero coupling does not require any switches to be activated. Similarly, the $N_2 = 1$ case [Fig. 8(b)] does not require any holograms, since it corresponds to a straightthrough interconnection. The $N_2 = 2$ case [Fig. 8(c)] requires the recording of two holograms, one for $N_1 = 0$ and 1, the other one for $N_1 = 2$ and 3. Finally, the $N_2 = 3$ case [Fig. 8(d)] requires one hologram, since it corresponds to an onto mapping. Therefore, the whole operation of residue multiplication modulo 4 can be implemented by $3 + 2 + 1 = 6$ holograms.

In general, the number of required holograms for multiplication mod $m = p^n$, where p is a prime number and n is a positive integer, can be obtained from

$$N_h = (n + 1)p^n - np^{n-1} - 2. \qquad (2)$$

The derivation of the above formula is provided in Appendix A. For the special case of $n = 1$, Eq. (2) is reduced to $N_h = 2p - 3$.

As an illustrative example, the number of required holograms for implementing residue multiplication moduli 3, 5, 7, 8 ($= 2^3$), 11, and 13 are 3, 7, 11, 18, 19, and 23, respectively. The 16-bit fixed-point multiplication that uses the above moduli can, therefore, be implemented by eighty-one holograms. This is about twice the corresponding number for 16-bit addition. The number of required channels and switches are the same as those for the addition case, i.e., forty-seven channels and 195 switches.

This method may be useful for some applications, but the problem is that the deflection of the laser beam to the appropriate hologram depends on both input numbers. This is sometimes practically difficult to achieve and requires a partial electronic processing. Also, since one of the numbers should be presented in two forms (as the input to a waveguide and as the input to the beam deflector) the system is not cascadable. The method described in the next subsection overcomes these shortcomings.

B. Increasing the Number of Channels

Another method for implementing residue multiplication is to increase the number of channels. In this method, the number of channels that are devoted to each value is determined by the maximum degeneracy of that value in the output. We demonstrate the procedure again by the residue multiplication modulo 4 case. As shown in Fig. 8, the maximum degeneracies of the values 0, 1, 2, and 3 in the output are 4, 1, 2, and 1, respectively. Therefore, a total of $4 + 1 + 2 + 1 = 8$ channels is needed to implement this operation (Fig. 9). The extra channels are used to make many-to-one mappings possible. In the input, only one channel is needed to code each value. The input values 0, 1, 2, and 3 are coded as the presence of light in the first, fifth, sixth, and eighth channel, leaving the other input channels idle. In the output, the presence of light in one of the first four channels is an indication of the result being equal to zero. If the result is 1, light should appear in the fifth channel; if it is 2, light should appear in either the sixth or seventh channel; and if it is 3, light should appear in the eighth channel.

Figure 9 shows how the required interconnections for residue multiplication modulo 4 can be obtained. The four cases shown in this figure correspond to $N_2 =$ 0, 1, 2, and 3, respectively. In each case, the switches that should be activated are marked. Notice that the $N_2 = 1$ case does not require activating any switches. Therefore, the whole process can be implemented by only three holograms. In general, using this technique, multiplication modulo m requires $m - 1$ holograms (same as the number required for residue addition).

The fact that more than one channel is devoted to some output values does not produce a problem in cascading these processors. One method for cascading is to merge all the output channels that correspond to a particular value by using a transition region. Also, notice that the above architecture used for implementing non-onto mappings can also be used to handle onto mappings. For example, the same waveguide couplers that are used for realizing residue multiplication can be used to realize residue addition as well. In this case, some of the channels will not be used, since for implementing an onto mapping, only one channel is needed for each value.

The number of channels required in this technique depends on the modulus. In general (see Appendix B), the number of required channels N_c for multiplication mod $m = p^n$, where p is a prime number and n is a positive integer, is given by

$$N_c = (n + 1)p^n - np^{n-1}. \qquad (3)$$

For the special case, where $n = 1$, Eq. (3) is reduced to $N_c = 2p - 1$.

It is interesting to note that the number of channels in this method is very close to the number of holograms in method A. In fact, the two numbers differ by a constant of 2. This difference is due to the two interconnections (corresponding to the $N_1 = N_2 = 0$ and $N_2 = 1$ cases) that are realized by default in the first method. If two holograms are considered for these cases, the two numbers become identical.

Another interesting point is that, although the number of input and output channels in Fig. 9 is eight, only five interaction layers and seventeen switches are used, because not all permutations of the input channels are needed. For example, if $N_1 = 3$, the input light does not have to be coupled to the first three channels of the output. In general, the number of required interaction layers N_l for residue multiplication is

$$N_l = N_c - m + 1, \qquad (4)$$

where N_c is the number of channels and m is the modulus. Having the number of layers, the number of switches N_s can then be found from the corresponding expression for an array of N_c channels and N_l layers, i.e.,

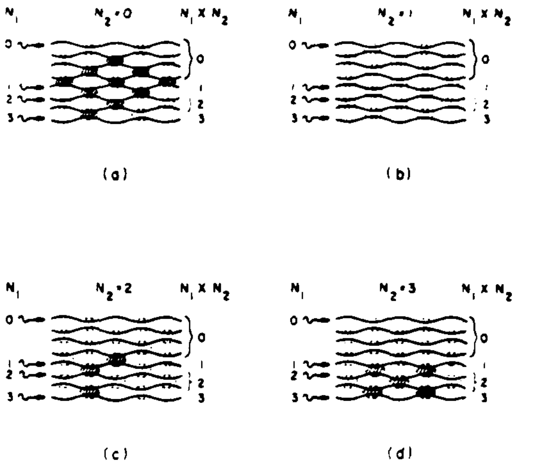


Fig. 9. Required switching states for implementing residue multiplication ($N_1 \times N_2$) modulo 4. The hatched switching elements are ON. The four interconnections realized in (a), (b), (c), and (d) correspond to $N_2 = 0, 1, 2$, and 3, respectively.

$$N_s = [(N_c - 1)N_l/2]. \qquad (5)$$

If N_c is even and N_l is odd, depending on the structure of the gate array, N_s is the nearest upper or lower integer of the value inside the brackets. It is possible to design the array so that N_s is the nearest lower integer.

As an illustrative example, the number of required channels for implementing residue multiplication moduli 3, 5, 7, 8 ($= 2^3$), 11, and 13 are 5, 9, 13, 20, 21, and 25, respectively. The 16-bit multiplication that uses the above moduli can, therefore, be implemented by ninety-three channels. The number of interaction layers required for the above moduli are 3, 5, 7, 13, 11, and 13, respectively. The corresponding number of switches are 6, 20, 42, 123, 110, and 156, which add up to 457. The number of required holograms is the same as the 16-bit addition case, i.e., forty-one.

VI. Other Applications

A major advantage of the second architecture is that it is cascadable. The output of the processor appears as the presence of light in a particular position, where an input channel of the next processor may exist. One possible application of the cascading property is the evaluation of polynomials. Horner's rule for polynomial evaluation is well known. For example,

$$\begin{aligned} P(x) &= a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0 \\ &= \{[(a_4x + a_3)x + a_2]x + a_1\}x + a_0. \qquad (6) \end{aligned}$$

This can be easily pipelined into a set of operations on an optical input signal using the values of x and a_i as the inputs to the deflectors (Fig. 10). Since positional coding has been used for data representation, minor light losses do not prevent such cascading. Polynomial evaluation is a very powerful operation because many functions can be represented accurately by a polynomial.

The proposed architecture is not limited to performing a series of arithmetic operations. In fact, any mapping in the residue system can be performed by this processor. To allow for all possible mappings, m channels should be devoted to each possible value, where m is the modulus used. Thus an array of m^2 channels is required for the most general operation. The input number N_1 can be coded as the presence of light in one of the channels that correspond to its value. Depending on the mapping of interest, a particular hologram is selected by the second input number N_2. The reconstructed light activates some of the switches and couples the input light to one of the m output channels that correspond to the result.

VII. Conclusions

A residue arithmetic processor based on optical Fredkin gate arrays has been introduced. The processor consists of optical waveguide couplers and a page-oriented holographic memory. The components needed for the fabrication of this device can be achieved with the present technologies in integrated optics and holography. The device is insensitive to variation in phase or polarization of light, since positional coding is used for data representation and processing. And finally, the processor is cascadable.

Realization of residue functions and operations with this processor has been described. The implementations of residue addition and multiplication have been analyzed in detail. The implementation of residue addition is straightforward, since all the mappings are onto. Residue multiplication is more complex, since some of the required mappings are not onto.

Two methods have been described to realize nononto mappings with optical Fredkin gate arrays. One method is to increase the number of holograms without changing the number of channels. The second method, which appears to be more powerful, is to increase the number of channels without changing the number of holograms. The latter technique has the advantage that the addressing of the holographic memory is determined by one of the input numbers and, therefore, can be achieved by a 1-D deflector, such as an acoustooptic cell.

The proposed processor is not restricted to the basic arithmetic operations. It has been shown that more complex operations, such as polynomial evaluation and general mapping, can be implemented with this architecture.

Appendix A: Number of Holograms in Method A

In this Appendix, analytic expressions are derived for the number of holograms required for implementing residue multiplication using the method described in Sec. V.A. The two numbers involved in multiplication are N_1 and N_2, where N_1 is coded as the presence of light in a channel waveguide, while both N_1 and N_2 are used as the input to the deflecting system. The required mappings for an example case can be seen in Fig. 8.

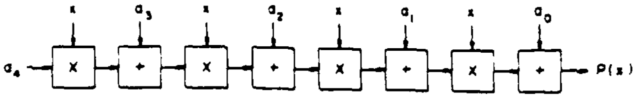


Fig. 10. Cascaded system for evaluating $P(x) = a_4x^4 + a_3x^3 + a_2x^2 + a_1x + a_0$ using Horner's rule.

If the modulus is a prime number (i.e., $m = p$), all the mappings, with the exception of the one that corresponds to the $N_2 = 0$ case, are onto. This special case requires $p - 1$ holograms, one for each of the nonzero values of N_1. No hologram is needed for $N_1 = N_2 = 0$, since zero-to-zero coupling does not require any switches to be activated. The remaining values of N_2 (i.e., $1, 2, \ldots,$ and $p - 1$) produce onto mappings. The $N_2 = 1$ case does not require any holograms, since it corresponds to a straightthrough interconnection. Each of the other cases requires one hologram. Therefore, the total number of holograms is $N_h = (p - 1) + (p - 2) = 2p - 3$.

If the modulus is not prime, more holograms are needed. The case of interest is when the modulus is a power of a prime number, i.e., $m = p^n$. The number of required holograms for this general case can be obtained by analyzing the mappings involved as follows:

(1) The $N_2 = 0$ case maps all the inputs to the zero output. Hence it requires $m - 1 = p^n - 1$ holograms.

(2) The $N_2 = kp$ cases, where $0 < k < p^{n-1}$ and k and p are relatively prime, map the inputs to the output ports that correspond to integer multiples of p. There are $(p - 1)p^{n-2}$ such cases, and each requires p holograms. Therefore, $(p - 1)p^{n-1}$ holograms are needed for these cases.

(3) In general, the $N_2 = kp^q$ cases, where $0 < q < n$, $0 < k < p^{n-q}$, and k and p are relatively prime, map the inputs to the output ports that correspond to integer multiples of p^q. There are $(p - 1)p^{n-q-1}$ such cases, and each requires p^q holograms. Therefore, to realize the cases corresponding to each value of q, the storage of $(p - 1)p^{n-}$ holograms is needed. Since q has $n - 1$ possible values, the total number of holograms corresponding to all $N_2 = kp^q$ cases is $(n - 1)(p - 1)p^{n-1}$. This includes the number of holograms obtained in (2).

(4) In all the cases considered so far, N_2 is an integer multiple of p. The total number of these cases is p^{n-1}. In the remaining $p^n - p^{n-1}$ cases, N_2 and p are relatively prime and produce onto mappings. The interconnection for one of these cases ($N_2 = 1$) can be realized without any hologram, while the others need one hologram each. Therefore, $p^n - p^{n-1} - 1$ holograms are needed to realize the interconnections corresponding to these cases.

The total number of holograms for implementing residue multiplication modulo $m = p^n$ using method A can then be obtained by adding the numbers derived in (1), (3), and (4). The result is

$$\begin{aligned} N_h &= (p^n - 1) + (n - 1)(p - 1)p^{n-1} + (p^n - p^{n-1} - 1) \\ &= (n + 1)p^n - np^{n-1} - 2. \end{aligned} \quad \text{(A1)}$$

Appendix B: Number of Channels in Method B

In this Appendix, analytic expressions are derived for the number of channels required for implementing residue multiplication described in Sec. V.B. The two numbers involved in multiplication are N_1 and N_2, where N_1 is coded as the presence of light in a channel waveguide, and N_2 is used as the input to the deflecting system. The required mappings for an example case can be seen in Fig. 8.

If the modulus is prime, except the mapping that corresponds to the $N_2 = 0$ case, the other mappings are onto. For this special case, all values of N_1 should be mapped to the zero output; hence p channels are required for the zero value. In all other cases, each output value has a degeneracy of one. Hence each of the values $1, 2, \ldots, p-1$ requires one channel. Therefore, the total number of required channels is $N_c = p + (p-1) = 2p - 1$.

If the modulus is not prime, more channels are needed. The case of interest is when the modulus is a power of a prime number, i.e., $m = p^n$. The number of required channels for this general case can be obtained by analyzing the maximum degeneracy of each output value as follows:

(1) The maximum degeneracy of zero in the output is p^n, which corresponds to the $N_2 = 0$ case. Therefore, p^n channels are needed for the zero value.

(2) Each value expressible as kp, where $0 < k < p^{n-1}$ and k and p are relatively prime, has a maximum degeneracy of p. There are $(p-1)p^{n-2}$ such values. Therefore, a total of $(p-1)p^{n-1}$ channels is needed for the above values.

(3) In general, each of the values expressible as kp^q, where $0 < q < n$, $0 < k < p^{n-q}$, and k and p are relatively prime, has a maximum degeneracy of p^q. There are $(p-1)p^{n-q-1}$ such values. Therefore, $(p-1)p^{n-1} = p^n - p^{n-1}$ channels are needed for each value of q. Since q has $n-1$ possible values, a total of $(n-1)(p^n - p^{n-1})$ channels is needed for all values expressible as kp^q. This includes the number of channels obtained in (2).

(4) All the values considered so far correspond to integer multiples of p. The total number of these cases is p^{n-1}. Each of the remaining $p^n - p^{n-1}$ values has a maximum degeneracy of one. Therefore, $p^n - p^{n-1}$ channels are needed for these values.

The total number of required channels for implementing residue multiplication modulo $m = p^n$ using method B can then be obtained by adding the numbers derived in (1), (3), and (4). The result is

$$\begin{aligned} N_c &= p^n + (n-1)(p^n - p^{n-1}) + (p^n - p^{n-1}) \\ &= (n+1)p^n - np^{n-1}. \end{aligned} \tag{B1}$$

References

1. For example, see these special issues on Digital Optical Computing: Proc. IEEE 72, No. 7 (1984); Appl. Opt. 25, Nos. 10, 14, 18 (1986); Opt. Eng. 24, No. 1 (1985); Opt. Eng. 25, No. 1 (1986); Opt. Eng. 26, No. 1 (1987).
2. A. W. Lohmann, "What Classical Optics Can Do for the Digital Optical Computer," Appl. Opt. 25, 1543 (1986).
3. J. W. Goodman, F. I. Leonberger, S.-Y. Kung, and R. A. Athale, "Optical Interconnections for VLSI Systems," Proc. IEEE 72, 850 (1984).
4. L. A. Bergman *et al.*, "Holographic Optical Interconnects for VLSI," Opt. Eng. 25, 1109 (1986).
5. J. Shamir, H. J. Caulfield, W. J. Miceli, and R. J. Seymour, "Optical Computing and the Fredkin Gates," Appl. Opt. 25, 1604 (1986).
6. J. Shamir and H. J. Caulfield, "High-Efficiency Rapidly Programmable Optical Interconnections," Appl. Opt. 26, 1032 (1987).
7. A. Svoboda and M. Valach, "Rational Numerical System for Residue Classes," in *Stroje na Zpracovani Informaci, Sbornik V.* (Nakl. CSAV, Prague, 1957), pp. 9–37 (in English).
8. N. S. Szabo and R. I. Tanaka, *Residue Arithmetic and Its Applications to Computer Technology* (McGraw-Hill, New York, 1967).
9. S. A. Collins, "Numerical Optical Data Processor," Proc. Soc. Photo-Opt. Instrum. Eng. 128, 313 (1977).
10. C. Y. Yen and S. A. Collins, "Operation of a Numerical Optical Digital Processor," Proc. Soc. Photo-Opt. Instrum. Eng. 232, 160 (1980).
11. C. C. Guest and T. K. Gaylord, "Truth-Table Look-Up Processing Utilizing Binary and Residue Arithmetic," Appl. Opt. 19, 1201 (1980).
12. M. M. Mirsalehi and T. K. Gaylord, "Truth-Table Look-Up Parallel Data Processing Using an Optical Content-Addressable Memory," Appl. Opt. 25, 2277 (1986).
13. A. Huang, "The Implementation of a Residue Arithmetic Unit via Optical and other Physical Phenomena," In *Proceedings, International Optical Computing Conference* (IEEE, New York, 1975), p. 14.
14. A. Huang, Y. Tsunoda, J. Goodman, and S. Ishihara, "Optical Computation Using Residue Arithmetic," Appl. Opt. 18, 149 (1979).
15. A. Tai, I. Cindrich, J. R. Fienup, and C. C. Aleksoff, "Optical Residue Arithmetic Computer with Programmable Computation Modules," Appl. Opt. 18, 2812 (1979).
16. D. Psaltis and D. Casasent, "Optical Residue Arithmetic: a Correlation Approach," Appl. Opt. 18, 163 (1979).
17. D. Psaltis, D. Casasent, D. Neft, and M. Carlotto, "Accurate Numerical Computation by Optical Convolution," Proc. Soc. Photo-Opt. Instrum. Eng. 232, 151 (1980).
18. S. Y. Huang and S. H. Lee, "Residue Arithmetic Circuit Design Based on Integrated Optics," Proc. Soc. Photo-Opt. Instrum. Eng. 321, 122 (1982).
19. For example, see R. Chen and C. S. Tsai, "Thermally Annealed Single-Mode Proton-Exchange Channel-Waveguide Cutoff Modulator," Opt. Lett. 11, 546 (1986).
20. For example, see J. W. Gladden and R. D. Leighty, "Recording Media," in *Handbook of Optical Holography*, H. J. Caulfield, Ed. (Academic, New York, 1979), pp. 277–298.
21. For example, see E. H. Young and S.-K. Yao, "Design Considerations for Acousto-Optic Devices," Proc. IEEE 69, 54 (1981).
22. C. C. Guest, M. M. Mirsalehi, and T. K. Gaylord, "Residue Number System Truth-Table Look-Up Processing—Moduli Selection and Logical Minimization," IEEE Trans. Comput. C-33, 927 (1984).
23. C. A. Papachristou, "The Recurrency Classes in Multi-Operand Addition and Multiplication Modulo M," in *Proceedings, 1981 Conference on Information Systems and Science* (John Hopkins U., Baltimore, Mar. 1981), pp. 402–407.

This work has been partially supported by the Office
of Naval Research under contract N00014-86-K-0591.

Pattern recognition using reduced information content filters

Joseph Shamir, H. John Caulfield, and Joseph Rosen

Pattern recognition by optical spatial filtering procedures is discussed using general considerations with the objective of reducing the information content in the spatial filter. The achievement of this objective is very useful toward the wide application of spatial light modulators and also for facilitating distortion invariant recognition. The proposed novel approach is demonstrated by an example employing bipolar spatial filters for rotation invariant pattern recognition.

I. Introduction

Usually the emphasis in research toward a useful optical pattern recognition architecture is the attainment of higher and narrower correlation peaks employing holographic spatial filters[1,2] with high information content. For real-time applications one would like to use devices like spatial light modulators that cannot handle these large amounts of information. The high information content is also a hindrance when distortion invariance such as rotation or scale change is considered. For example, both the matched filter[2] and its more recent variant, the phase-only matched filter,[3,4] yield high correlation peaks. Unfortunately, these filters are the most intolerant of any distortion because a large part of their information content is that of the orientation and scale of the object.

The main objective of this work is development of a pattern recognition approach taking into consideration the resolution limitations of presently available spatial light modulators. To achieve this goal we seek a procedure for reducing to a minimum the amount of information to be written on these modulators when they are employed in the input and filter planes of a pattern recognition system. It is evident that the penalty to be paid is a reduction in the quality of the correlation peaks, but this will be a suitable price for higher flexibility and easier applicability.

We start from general considerations that are independent of the particular architecture to be adopted. Most of the steps described may be applied to a diverse set of configurations. For example, they are valid for coherent or incoherent pattern recognition performed by employing spatial frequency filtering or template matching. To obtain shift invariance we shall restrict the discussion to spatial filtering procedures over the Fourier transform plane.

II. General Considerations

We define our goal to be the recognition of each pattern in a set of N patterns, $f_i(x,y)$, $(i = 1,2,\ldots,N)$. The limitation to N predetermined patterns is not so severe as it seems at first sight, since one or more of these patterns may be noise or background. We form 2-D Fourier transforms (FTs), $F_i(u,v)$, and wish to manufacture a set of filters $M_j(u,v)$, $(j = 1,2,\ldots,N)$ in such a manner that we obtain an optimal response represented schematically by the relation

$$\mathbf{R}_{ij} = \mathrm{O}[F_i(u,v); M_j(u,v)] = \delta_{ij}, \tag{1}$$

where O is some operator. The degree to which we can approach this ideal response depends on the operator, the set of filters, and the patterns involved. For example, we may consider the integral power reaching the output plane of the optical system, $O(x,y)$, indicated in the schematic representation of Fig. 1. By Parseval's theorem this power is identical with the power transmitted by the filter positioned at the FT plane [$M(u,v)$ in the figure]. For this configuration criterion (1) has the form

$$\mathbf{R}_{ij} = \int |F_i(u,v)M_j(u,v)|^2 dudv = \delta_{ij}. \tag{2}$$

This, however, is a paradoxical requirement since we deal with a positive definite integrand, and one may

Joseph Rosen is with Technion—Israel Institute of Technology, Department of Electrical Engineering, Haifa 32000, Israel; the other authors are with University of Alabama in Huntsville, Center for Applied Optics, Huntsville, Alabama 35899.

Received 1 December 1986.
0003-6935/87/122311-04$02.00/0.

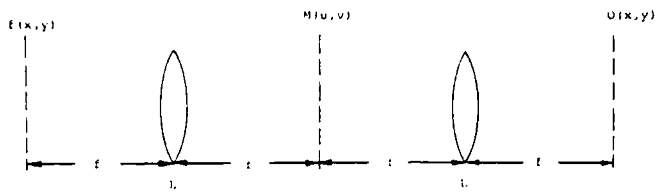


Fig. 1. Spatial filtering system: L, Fourier transforming lenses of focal length f; $f(x,y)$, input pattern; $O(x,y)$, output pattern; and $M(u,v)$, filter function.

have a nonvanishing filter function only for $i = j$. Naturally, such a criterion cannot lead to a selective set of filters, and one should seek a solution that involves the analysis of a power redistribution over the output plane.

As our starting point we refer to Fig. 1 and define the response according to Eq. (1) as the power incident at the origin of the output plane. (Since we are dealing with Fourier plane filtering the position of this origin corresponds to the position of the object in the input plane.) Denoting by $O_{ij}(x,y)$ the output distribution produced by pattern $F_i(u,v)$ illuminating filter $M_j(u,v)$ this recognition criterion states

$$\mathbf{R}_{ij} = |O_{ij}(0,0)|^2 = \delta_{ij}, \tag{3}$$

where, in the configuration of Fig. 1,

$$O_{ij}(x,y) = \mathcal{F}[F_i(u,v)M_j(u,v)], \tag{4}$$

and Eq. (1) is now equivalent to

$$|\int F_i(u,v)M_j(u,v)dudv| = \delta_{ij}. \tag{5}$$

This relation represents a set of linear equations that can be solved, at least in principle, to generate the filters $M_j(u,v)$.

III. Filter Generation

To solve Eq. (5) for each filter and generate M_j we have to sample the Fourier plane. Assuming a rectangular coordinate system we divide the Fourier plane into $K \times L$ regions of area s_{kl}, each (not necessarily equal) with $k = 1,2,\ldots ,K$ and $l = 1,2,\ldots ,L$. To each of these regions we designate a constant value M_{jkl} as its (generally complex) amplitude transmittance.

Integrating the incident complex amplitude over each region we form the matrix elements

$$F_{ikl} = \int_{s_{kl}} F_i(u,v)dudv, \tag{6}$$

and we may generate the filter samples by solving the set of N^2 linear equations:

$$\left|\sum_{k=1}^{K}\sum_{l=1}^{L} F_{ikl}M_{jkl}\right| = \delta_{ij}, \tag{7}$$

where $i, j = 1,2,\ldots ,N$.

Equation (7) gives N equations for each of the N filters $M_j(u,v)$ consisting of $K \times L$ unknown samples. Thus one may obtain a unique solution if $K \times L = N$. This is a very far reaching consequence as it means that to discriminate among N patterns it is adequate to use filters with N transmittance values. We have to point out, however, that the above conclusion is only theoretical and holds if filters and detection can be implemented with infinite dynamic range and infinite accuracy. Furthermore, the above relations were obtained by constraints imposed on a single point in the output plane. For a satisfactory discrimination, taking into account practical considerations, this will usually not be adequate, and the number of equations (and samples) will have to be multiplied by the number of required discriminating points. This procedure essentially generates a synthetic discriminant function (SDF).[5]

We considered up to this point $N \times L$ rectangular sample regions just as an example. To attain efficient recognition the area and shape of these samples must be optimized according to the recognition task. For another example we consider rotation invariant pattern recognition with rotationally invariant filters. For this case the filter division is along concentric rings. Denoting the radius of the kth ring by r_k we may have to look for an optimal function $h(k)$ that gives the various radia

$$r_k = h(k). \tag{8}$$

An interesting and simple class of these functions can be written in the form

$$h(k) = r_1 k^q, \tag{9}$$

where r_1 and q are constants. The special case of $q = 1/2$ is the Fresnel zone division where all the rings have the same area, while the case $q = -1/2$ may be termed the inverse Fresnel zone plate (i.e., the kth radius of the Fresnel zone plate multiplied by the kth radius of the inverse Fresnel zone plate is a constant for all k). These two kinds of division complement each other with respect to the nature of patterns to be discriminated. The first kind of division has rings that become very narrow for high spatial frequency values, thus making it a good rotation invariant filter for patterns having their important features at high frequencies. Conversely, the second choice will be suitable for filtering information at low spatial frequencies. An intermediate case may be treated with filters having $q = 1$ where the width of the rings is constant. This analysis is reminiscent of the procedures utilized in Ref. 6 where a specific circular harmonic was chosen for each recognition task depending on the objects to be dealt with. Sometimes the useful information is concentrated only in certain regions of the filter plane. For example, in many cases the low frequency region does not contain selective information, and better filtering is obtained by eliminating the energy in this region altogether.

A similar procedure would be implemented for complete scale invariant pattern recognition where the filter should depend on angular orientation only and not on the distance from the origin. For this case one would need radial division lines to split the filter plane into L sectors.

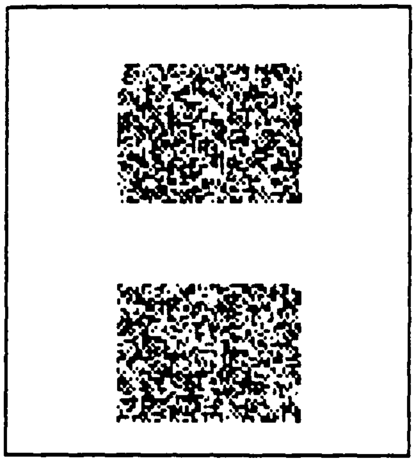


Fig. 2. Two random patterns to be discriminated.

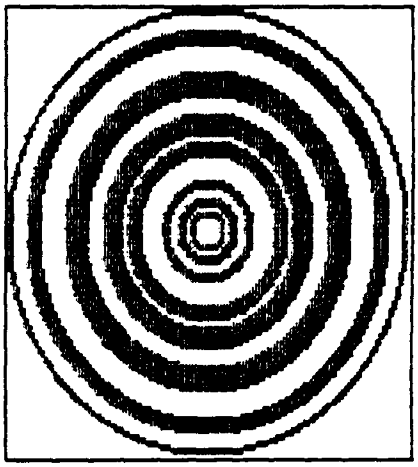


Fig. 3. Rough representation of a rotationally invariant bipolar filter made for recognizing the top pattern of Fig. 2.

IV. Bipolar Filters and Experiment

In principle the filters described in this work can be generated similarly to other composite filters[7] or circular harmonic filters[6] as computer generated holograms. However, the present procedure has a more general attitude, and other implementations are also possible. Although the information content of these filters is relatively low, a holographic filter needs still a quite large bandwidth. To reduce this requirement we show now that filters with real, positive, and negative valued transmission characteristics can perform reasonably well even for rotation invariant pattern recognition. It has been shown[8] that the implementation of such bipolar filters is possible, and with the advent of spatial light phase modulators the procedure becomes rather simple. One major advantage of working with nonholographic spatial filters is the in-line architecture of the whole optical system.

In a bipolar filter the amplitude transmittance of each filter element is real and satisfies the relation

$$-1 \le M_{jkl} \le 1. \qquad (10)$$

This is a very serious constraint on the equations determining these values [Eq. (7)], and in many cases such solutions are not available. The only way to get around this problem is to relax the conditions on the right-hand side of the equations and optimize the solutions.

To demonstrate the procedure we implement a completely rotation invariant filter. For a general treatment of rotation or scale invariant pattern recognition, it is useful to represent the input pattern in polar coordinates. We denote by $F(r,\theta)$ the complex amplitude distribution produced by the input pattern at the filter plane, and we employ a circularly symmetric filter. We divide the filter plane into N concentric rings (where N is now the total number of divisions as discussed in the previous section) and denote by M_{jk} the transmittance (real, positive, or negative) of the kth ring in the jth filter. Equation (6) can be now rewritten in the form

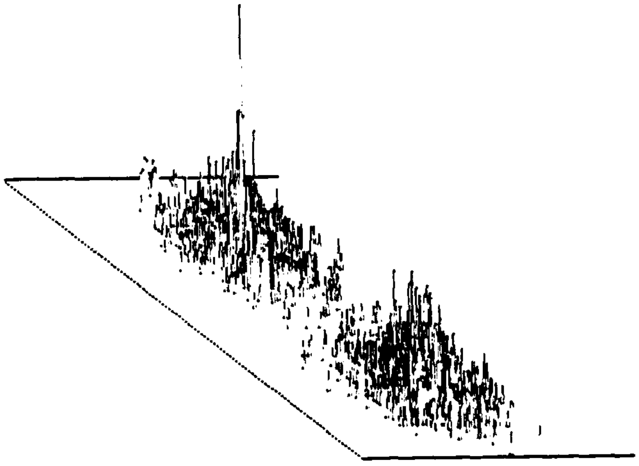


Fig. 4. Output intensity distribution with input of Fig. 2 and operation with the filter of Fig. 3.

$$F_{ik} = \int_{s_k} F_i(r,\theta) 2\pi r dr d\theta. \qquad (11)$$

where integration is performed over the area of the kth ring s_k. With these definitions Eq. (7) will be replaced by

$$\left| \sum_k F_{ik} M_{jk} \right| = \delta_{ij}. \qquad (12)$$

Since this relation concerns the absolute values of each equation, an arbitrary phase may be assigned to render the values of M_{jk} real.

To test the viability of the present approach some computer experiments were performed, and rotation invariant recognition was demonstrated. One experiment involved random patterns as shown in Fig. 2. The filter plane was divided into sixty-four concentric rings, and filters were generated according to Eq. (12). Figure 3 is an approximate representation of the rotationally invariant filter made for one of the patterns, while Fig. 4 is the intensity distribution over the fil-

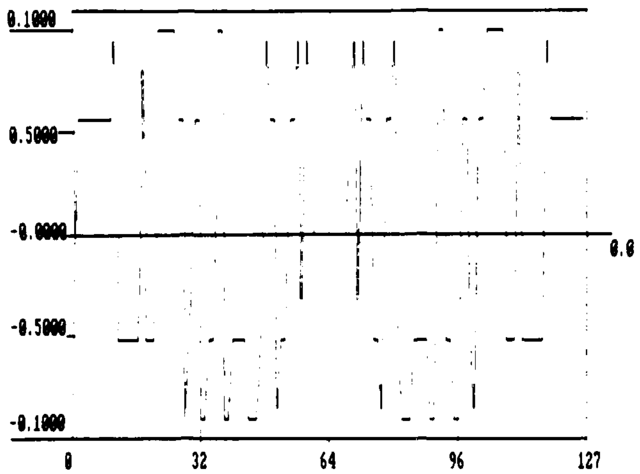


Fig. 5. Cross section along a diameter of the filter with removal of low frequency components.

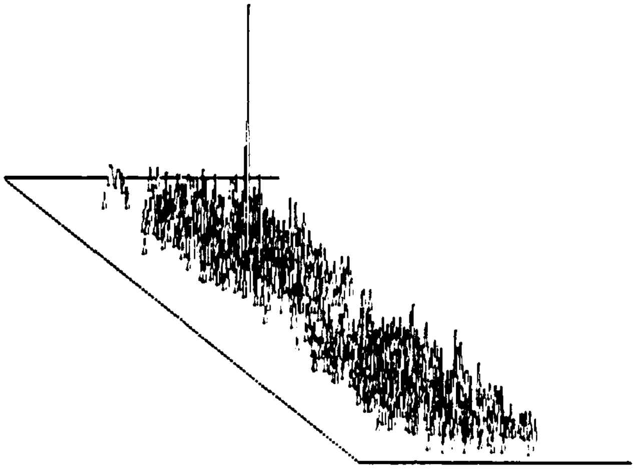


Fig. 6. Output intensity distribution for the input of Fig. 2 and filter of Fig. 5 prepared for recognizing the top pattern.

tered output plane. The result is quite noisy in part due to a large fraction of energy transmitted at zero spatial frequency that contains no information about the object. If this frequency component is removed by a modified filter, the cross section of which is shown in Fig. 5, the filtered output shown in Fig. 6 is obtained with an appreciably enhanced SNR.

V. Conclusions

A simplified approach to optical pattern recognition was proposed to make its practical application more feasible. As an example of possible implementation of the present approach a recognition criterion was chosen so that the filters contain information about the complete complex amplitude distribution of the patterns. Using computer experiments it was shown that adequate information may be contained in bipolar filters to recognize patterns even in a completely shift and rotation invariant manner with no need for holographic filters. In a subsequent publication it will be shown that the approach presented here can be employed for different kinds of filter, i.e., phase filters, and patterns of various nature.

It should be emphasized that criterion (1) can never be exactly satisfied. Further studies are carried out to search for possibly better criteria that may also be easier to implement optically.

This work was partially supported by a contract with NASA Johnson Space Center under contract NAS9-27598.

References

1. See, for example, S. P. Almeida and G. Indebetouw, "Pattern Recognition via Complex Spatial Filtering," in *Applications of Optical Fourier Transforms*, H. Stark, Ed. (Academic, Orlando, 1982).
2. A. B. VanderLugt, "Signal Detection by Complex Spatial Filtering," IEEE Trans. Inf. Theory **IT-10**, 139 (1964).
3. H. J. Caulfield, "Role of the Horner Efficiency in the Optimization of Spatial Filters for Optical Pattern Recognition," Appl. Opt. **21**, 4349 (1982).
4. J. L. Horner and P. D. Gianino, "Applying the Phase-Only Filter Concept to the Synthetic Discriminant Function Correlation Filter," Appl. Opt. **24**, 851 (1985).
5. D. Casasent and W.-T. Chang, "Correlation Synthetic Discriminant Functions," Appl. Opt. **25**, 2343 (1986).
6. H. H. Arsenault and Y. Sheng, "Properties of Circular Harmonic Expansion for Rotation-Invariant Pattern Recognition," Appl. Opt. **25**, 3225 (1986).
7. H. J. Caulfield and W. T. Maloney, "Improved Discrimination in Optical Character Recognition," Appl. Opt. **8**, 2354 (1969).
8. A. Furman and D. Casasent, "Bipolar Incoherent Optical Pattern Recognition by Carrier Encoding," Appl. Opt. **18**, 660 (1979).

Invited Paper

Rotation-Invariant Pattern Recognition and Some of Its Limitations

H. J. Caulfield and Joseph Shamir

Center for Applied Optics, The University of Alabama in Huntsville
Huntsville, Alabama 35899

Abstract

Rotation-invariant pattern recognition is shown to have intrinsic limitations determined by the set of patterns to be recognized and by the specific optical setup. Within these limitations, a general procedure is proposed for the generation of bipolar filters that do not require the sensitive alignment procedures involved in holographic filters and are suitable for superposition synthesis to achieve rotation invariance.

Introduction

The oldest and most straight forward approach to pattern recognition is image plane analysis (or template matching) where one compares the image of the object with some stored pattern. If the object is, for example, a typed page, one should scan the page to locate each letter and then compare it with given template letters. To consider the additional possibility that some of the letters may be rotated we shall have to perform a rotation for each template at each letter position. In an automatic system that has to perform all these operations, we are confronted by an incredibly time consuming task even for the most advanced computers. Therefore it would be very useful to replace the templates by some rotation-invariant filters whenever possible.

Besides the problem of rotation, the main drawback of image-plane analysis is its position dependence. This problem is resolved by transferring the image plane operation to the Fourier-plane where the whole input information is addressed simultaneously. The conventional optical system perform this procedure is shown in Fig. 1. A coherent plane wave illuminating the input function f(x,y) operates its Fourier transform (FT) over plane M giving rise to a complex amplitude distribution, F(u,v). One may record the intensity distribution on a photographic plate to produce a filter with amplitude transmittance $|F^2|$. Reinserting this filter into place M and replacing the function f by some other function, g(x,y) produces a complex amplitude distribution immediately to the right of M given by $G|F|^2$, where g is the FT of g. An additional FT performed by the second lens yields, on the output plane, O, the triple convolution:

$$g(x,Y) * f^*(-x,-Y) * f(x,Y) \tag{1}$$

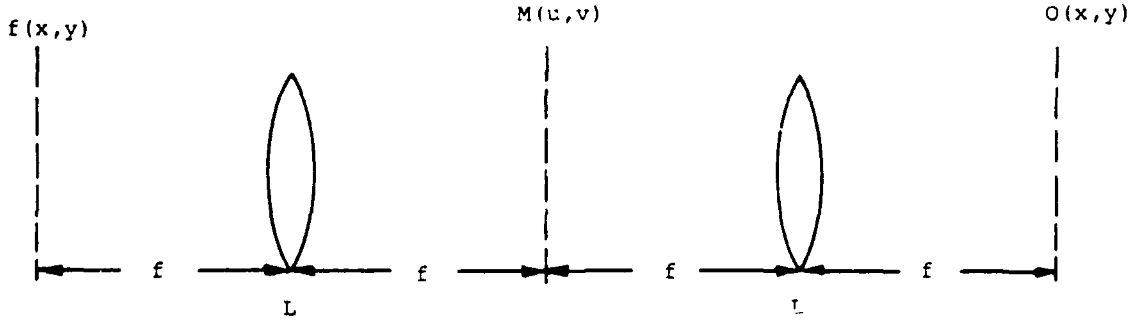


Fig. 1. In-line spatial filtering system: L-Fourier transforming lenses of focal length f; f(x,y) - input pattern, O(x,y) - output pattern and M(u,v) - filter function.

where * denotes convolution and f^* is the complex conjugate of f. Expression (1) is the required cross-correlation of f and g but this is convolved with the function f that makes it a rather poor measure of correlation even before considering rotations. It should be noted, however, that this kind of filter is just one possibility. For example, a better response could be obtained by using the same filter as an intensity filter instead of amplitude filter with incoherent illumination[1].

To obtain an appreciable improvement over the above considered possibilities, most of the presently practiced optical methods for pattern recognition [2] are based on the holographic matched filter, first proposed by VanderLugt[2]. This Fourier-plane filter has a high resolution but is very sensitive to misalignment, object scaling and object rotation. From the practical point of view one seldom needs this high sensitivity and the stringent alignment requirements limit the applicability of method.

Many attempts were made to render the VanderLugt filter orientation insensitive: Various averaging methods [4-12] to produce filters that recognize classes of objects rather than specific patterns were quite successful and, to some extent, could also be generalized to treat a range of angular orientations. Spatial multiplexing techniques[13-15] are useful in principle but not very practical due to alignment problems and, in most cases, the involvement of mechanically moving components. Photodetector array detection on the Fourier plane[16] and computer processing is also possible but it is applicable only for single-pattern-at-a-time analysis. Recently some more sophisticated rotation-invariant methods were proposed where the spatial filters are based on circular harmonic decomposition generated by digital computers and recorded holographically[17-18] In principle this approach proved quite efficient but, unfortunately, it involves elaborate hardware and the use of inconvenient components such as liquid gates that hinder its practical application. The generalized analysis of Ref. 19 may be helpful to estimate the response of filters with various degress of rotation invariance to a specific input but this should be augmented with some derivation of filter selectivity to different inputs.

In this work we address first the general question of the limitations imposed on rotation invariant pattern recognition by the intrinsic nature of optical methods. It will be indicated that the answer depends on the specific patterns to be recognized and on the actual procedures applied. With these limitations kept in mind we propose a new approach to the synthesis of filters. This approach should be straightforward to implement and easy to use in practice.

II. Some Limitations On Rotation-Invariant Pattern Recognition

Addressing the general question of rotation-invariant pattern recognition, we use polar coordinates to represent the input pattern, $f(r,\theta)$. At this state, f represents the complex amplitude distribution produced by the input pattern at the filter plane where we insert a filter with amplitude trasmittance $m(r,\theta)$. This plane may be either the image plane or the Fourier plane, whichever is more convenient for a certain application. As pointed out in Ref. 19 there are a number of ways to define the performance of a filter. One of these possibilities is the integral detection of all the light arriving at the output plane. Using the principle of energy conservation this integral quantity is given by the total power transmitted by the filter:

$$R(0) = \int_0^{2\pi}\int_0^{r_m} \left| m(r,\theta)f(r,\theta) \right|^2 rdrd\theta \qquad (2)$$

where r_m represents the size of the filter assumed circular. To investigate the response for rotated objects we may keep the object constant and rotate the filter assuming that all the rest of the system is circularly symmetric. The response with the filter rotated into the θ_0 orientation may be described by the relation,

$$R(\theta_0) = \int_0^{2\pi}\int_0^{r_m} \left| m(r,\theta - \theta_0) \right|^2 \left| f(r,\theta) \right|^2 rdrd\theta \qquad (2a)$$

If we want to make this response rotation-invariant we have to require,

$$\frac{\partial R(\theta)}{\partial \theta_0} = 0 \qquad (3)$$

which leads to the obvious result that $|m|$ should be independent of the angular coordinate, θ apart from a phase variation that may change the output distribution but not its integral power. Thus, to generate a rotation-invariant filter one has to apply some amplitude averaging procedures over the angular coordinate.

The most general result from these considerations is that rotation invariant pattern distinction is possible only among patterns the angular average of which differ from each other at the filter plane. Later it will be indicated that the response of Eq. 2 is not very discriminant detection. Nevertheless, the conclusion regarding the angular independence of $|m|$ for rotation invariance is quite general but in most cases restrictions may arise also for the phase variation.

It is very useful to note here that the same class of patterns that is suitable for rotation-invariant recognition in the image plane may be impractical for rotation invariant recognition in the Fourier plane. Difficulties that may occur can be illustrated by consideraing the simple block characters as shown in Fig. 2. Assuming that the lines composing the characters are transparent while all the rest is opaque, we rotate each character around its center of mass and record the transmitted intensity in the image plane. Many of the patterns $m(r)$ generated this way will have different features characteristics of the original letter. Thus, in principle, these masks may serve as some crude rotation invariant recognition "templates" for the set of characters. This may not be the case if we convert to Fourier plane analysis.

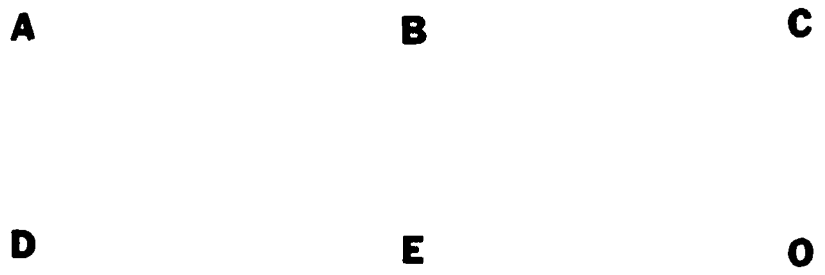


Figure 2. A sample of characters for recognition.

The most general result from these considerations is that rotation invariant pattern distinction is possible only among patterns the angular average of which differ from each other at the filter plane. Later it will be indicated that the response of Eq. 2 is not very discriminant detection. Nevertheless, the conclusion regarding the angular independence of $|m|$ for rotation invariance is quite general but in most cases restrictions may arise also for the phase variation.

It is very useful to note here that the same class of patterns that is suitable for rotation-invariant recognition in the image plane may be impractical for rotation invariant recognition in the Fourier plane. Difficulties that may occur can be illustrated by consideraing the simple block characters as shown in Fig. 2. Assuming that the lines composing the characters are transparent while all the rest is opaque, we rotate each character around its center of mass and record the transmitted intensity in the image plane. Many of the patterns m(r) generated this way will have different features characteristics of the original letter. Thus, in principle, these masks may serve as some crude rotation invariant recognition "templates" for the set of characters. This may not be the case if we convert to Fourier plane analysis.

Figure 3 shows the optically generated FT of the characters in Fig. 2. The highest and most intense spatial frequency components are generated by the lines constructing the characters. The absolute magnitude of these components is almost identical for all the characters, the distinguishing feature being only in the orientation of the FT produced by the various line segments. At first sight it would appear that a rotationally invariant mask placed in this plane will have difficulties in distinguishing among these unless it can resolve minute differences due to various lengths of the line segments. It can be seen, however, from the rotational averages whown in Fig. 4 that appreciable differences still exist but they will decrease as the line segments get longer when compared to their width. Stated in a more general way, the class of patterns generated by narrow lines (such as line drawings) where the line-width is the minimum feature size with all other features having much larger dimensions, is not suitable for rotation-invariant pattern recognition in Fourier plane procedures.

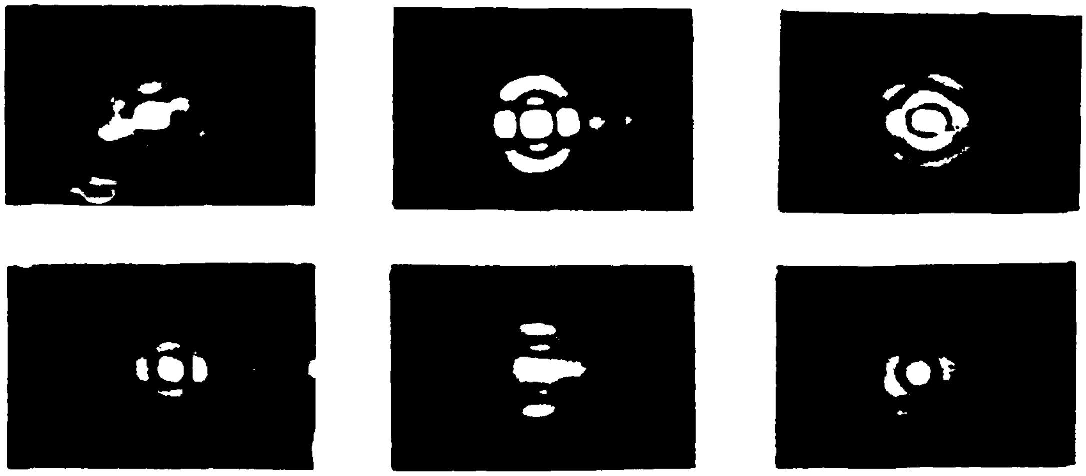


Figure 3. Optical Fourier transform of characters from Fig. 2.

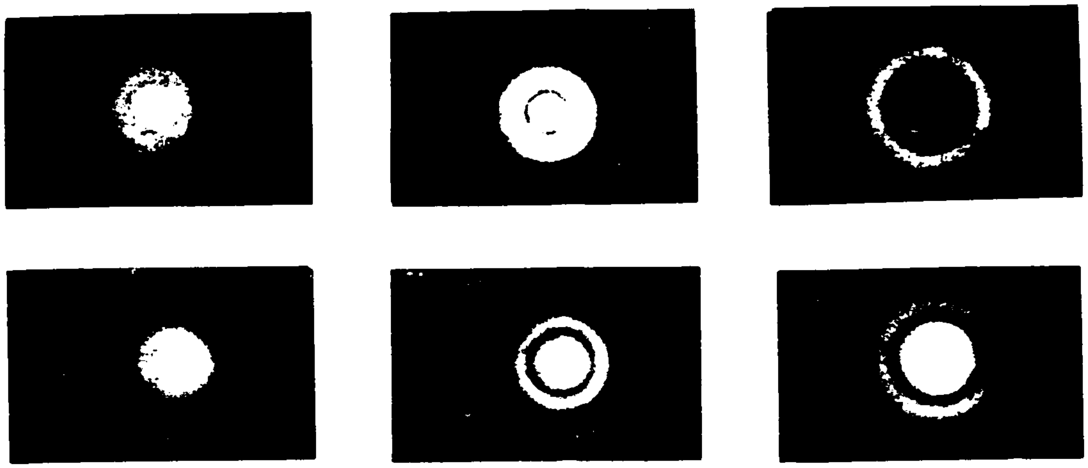


Figure 4. Rotation averages of transforms in Figure 3.

III. Bipolar Filters For Pattern Recognition

The two extreme approaches for the generation of spatial filters for pattern recognition, i.e. the power spectrum filter and the holographic matched filter, were discussed in the introduction. Observing the present state of art it appears that the performance of the first kind is too poor while the applications of holographic filters are tedious and frequently require quite sophisticated hardware and software. In this section we introduce a modified approach to filter synthesis that should be simple to implement and to use. To achieve this goal we consider the following requirements: a) The spatial filter should be less sensitive than a holographic filter but should have a better performance than the simple power spectrum filter; b) The pattern recognition system should give a high response for a specific input together with a negligible output for any other pattern included in a given set; c) Since we are interested in rotation-invariant pattern recognition the proposed method should be applicable for this purpose too. One way to meet requirement (a) is by the synthesis of a medium resolution filter that discards most of the phase information retaining some of it in a bipolar form. The filter will also conform with requirement (b) if its generation will take into consideration all the patterns to be analyzed. In the following we propose a number of variations to the implementation of such filters and convert them into rotation-invariant filters in the next section.

Bipolar pattern recognition is usualy considered in connection with incoherent illumination[1,20]. However in the present procedure we start from general considerations that are applicable to incoherent as well as coherent illumination, either for processing performed in the image plane or the Fourier plane. To keep this work within reasonable limits we shall mainly address Fourier plane processing with coherent illumination.

In Refs. 4-8 various mathematical procedures were investigated for the synthesis of composite matched spatial filters. Adopting a similar approach we rely on the fact that the procedure has been proven to be mathematically sound and simplify the derivation by avoiding steps such as decomposition into sets of orthor-normal functions. Consider a set of patterns, $f_i(x,y)$, $(i = 1,2,\ldots N)$ to be discriminated against each other. We form their 2-dimensional FT, $F_i\ (u,v)$ and wish to manufacture a set of filters $M_j\ (u,v)$, $(j = 1,2,\ldots N)$ in such a way that they will transmit light only if illuminated by a specific pattern. Mathematically, this requiremwent may be expressed by the relation.

$$\iint F_i\ (u,v)\ |^2\ |\ M_j(u,v)^2 dudv = \delta_{ij}.$$

where we assume M_j to be a complex amplitude transmission function. Unfortunately this is a paradoxical requirement since we deal with positive definite integrands. Therefore, any $|F_i| \ddagger 0$ requires $|M_j\ | = 0$ for all $i \ddagger j$ resulting with, at the most, one filter that transmits light. Naturally, such a criterion cannot lead to a selective set of filters indicating that the response function of Eq. (2) is not a good choice for our purposes. To improve performance we take one step towards a holographic matched filter (the main function of which is a certain redistribution of the power over the output space) and try a bipolar set of filters with the requirement,

$$\iint |\ F_i(u,v)\ M_j(u,v)\ dudv = \delta_{ij}. \qquad (4)$$

One way to synthesize a set of real filters that satisfy this relation is by the linear superposition (a composite filter).

$$M_j = a_{kj} \mid F_k(u,v) \mid \qquad (5)$$

where a_{kj} are real (positive or negative) constants and summation over identical indices is postulated. Although the functions $\mid F_i\ (u,v) \mid$ do not constitute an orthonormal set one may still substitute Eq. (5) into Eq. (4) and solve the following set of lequations,

$$a_{kj} d \iint \mid F_i\ (u,v) \mid \mid F_k(u,v) \mid \quad dudv = \delta_{ij} \qquad (6)$$

to evaluate a_{kj}.

Defering to a later stage the discussion of some difficulties involved in this procedure, we proceed to investigate the performance of the system asuming that we possess a filter set described by the characteristics indicated in the above relations. Inserting one of the filters, M_j, into the Fourier plane, M of Fig. 1, we illuminate it with pattern f_n, placed in the input plane. For convenience, we write the FT of f_n in the form,

$$F_n(u,v) = \mid F_n(u,v) \mid \exp i\psi_n(u,v) \qquad (7)$$

where ψ_n (u,v) is a real phase function. The complex amplitude distribution Eq. (7) is transmitted by the filter and transformed by the second lens to produce the output distribution,

$$O(x,y) = \mathfrak{F}(F_n M_j) = a_{kj} \mathfrak{F}(\mid F_n \mid \mid F_k \mid \exp i\Psi_n) \qquad (8)$$

where $\mathfrak{F}$ represents FT. In most cases of interest here, where we shall deal with real input patterns, the main contribution of the phase function will be a translation of the output pattern to a location corresponding to its position on the input place. Thus it is useful to consider alternative form:

$$O(x,y) = a_{kj} \mathfrak{F}(\mid F_n \mid \mid F_k \mid) * \mathfrak{F}(\exp i\ \psi_n) \qquad (9)$$

Since convolution is a linear process, one may perform the summation on k first and the convolution afterwards. The summed function reduces at the origin to Eq. (6). Thus one would expect a strong correlation for n = k with some weak and blurred responses for all other input patterns produced by some contribution away from the origin. The convolution with the phase function will introduce a partial reconstruction of the object and position it according to its location in the input plane.

IV. Rotation-Invariant Spatial Filters

In the following, we restrict ourselves to sets of patterns that are, in principle, suitable for rotation invariant recognition and use a procedure similar to the previous one.

We start by representing the functions, F_i in a polar coordinate system,

$$F_i(x,y) = F_i\ (r,0) \qquad (10)$$

and define a set of positive, real, normalized and rotation-invariant functions,

$$\underline{F}_i(r) = \frac{\mid F_i(r,\theta) \mid d\theta}{\mid F_i(r,\theta) \mid 2rdrd\theta} \qquad (11)$$

According to our discussion on the limitations with rotation invariant pattern recognition we would expect that discrimination will be possible if these functions differ "appreciably" among themselves. The amount of difference implied by the word "appreciably" depends on the actual experimental systems. The differences may be very minute for computer simmulation with arbitrary accuracy but should be much larger for a real system. We shall return to this point at the end of this section.

Following our previous procedure, we search for a set of filters that satisfy the orthogonality relation,

$$\int f_i(r)M_j(r)2\pi r dr = \delta_{ij} \tag{12}$$

Here too, we shall generate these filters as linear combination of the input patterns:

$$M_j(r) = a_{kj}\underline{F}_i\ (r) \tag{13}$$

Substitution into Eq. (12) leads again to N linear algebraic equations

$$a_{kj}\int \underline{F}_i(r)\underline{F}_k(r)2\pi r dr = \delta_{ij} \tag{14}$$

To implement this filter set, one should determine the set $\underline{F}_i$, solve the linear equations for $a_{kj}$and construct the filters accordingly.

Since the set $\underline{F}_i$ is positive definite, the solution of Eq. (14) will lead to positive and negative values for a_{kj}. As mentioned earlier, we gave up the complete phase dependence of the holographic filter but we still require bipolar values.

There are a number of possible approaches for the practical construction of the above described filters and here we proceed to describe one that may be called a quantized filter. Recalling that a rotation-invariant filter should be circularly symmetric, we divide the Fourier plane into N concentric rings. (The spacing of these rings and the number N will be discussed in the next section). We now represent each of the N normalized functions, $\underline{F}_i(r)$ (Eq. 11) by a vector P_i, with components p_{ij}, proportional to the square root of the total power incident on the j-th ring. The N circularly symmetric filters will be also represented by vectors in N-space, M_k, with components m_{kj}, that are the amplitude transmittance of the j-th ring in the k-th filter. This circular filters will attain its proper function if we require again the orthogonality relation,

$$p_{ij}m_{kj} = \delta_{ik} \tag{15}$$

This equation constitutes N^2 linear equations for the determination of the N^2 elements of the matrix m_{kj}. Since we are dealing here with a matrix it will be convenient to put the whole procedure into a matrix form: We construct the two matrices,

$$P = \begin{array}{l} p11\ p12\ \ldots\ p1N \\ p21\ p22\ \ldots\ p2N \\ \vdots \\ p_{n1}\ \cdots\cdots\cdots\ p_{NN} \end{array} \tag{16}$$

and

$$\begin{array}{l} m_{11}\ m_{21}\ \cdots\ m \\ m12\ m22\ \ldots\ \quad N1 \\ \vdots \\ m_{1N}\ \cdots\cdots\cdots\ m_{NN} \end{array} \tag{17}$$

and write Eq. (15) in the convenient form,

$$P\ M = I \tag{18}$$

where I is the unit matrix. Thus we see that the calculation of our filter reduces to a simple inversion of a matrix derived from measurement during the learning stage

$$M = P^{-1} \tag{18a}$$

Equations (6) and (14) may be also put in this matrix form by replacing m_{kj} and defining.

$$P_{ik} = \iint |F_i\ (u,v)|\,|F_k(u,v)\ dudv \tag{19}$$

for Eq. (6) and similarly for Eq. (14), using the normalized functions It is interesting to note that Eq (18) represents a symmetric matrix which is not necessarily the case for the original matrix defined in Eq. (15).

The limitations in the implementation of rotation-invariant filters discussed in section II are implicitly included in eq. (18a) and we may return to the phrase, "appreciably different" mentioned at the beginning of this section. It is immediately evident that if we have two identical patterns the matrix cannot be inverted. However, if there are two patterns that differ only slightly, a solution may be obtained but with $|m_{jk}|_{max}$. The result is that equations (in EQ. 15) having zero on their right hand side will not their right hand side will not be affected but, unfortunately, many of the values unity will have to be divided by the values unuity will have to be divided 1 we shall have to deal with an autocorrelation represented by a small fraction $1\ |\ m_k\ |_{max}$ that may be undetectable. For any practical system, this number will determine the limits imposed on recognition possibilities.

V. Filter-Place Division

In the previous section the filter plane ws divided into N concentric rings without specifying their widths. The reason is that the optimal division actually depends on the class of patterns to be recognized In principle one may use an arbitrary function to derive the radius r_n, of the n-th circle in the filter plane,

$$r_n = h(n)$$

An interesting and simple class of these functions can be written in the form,

$$h(n) = r_1 n^q \tag{21}$$

The special case of q = 1/2 is the Fresnel Zone division where all the rings have the same area while the case q = -1/2 may be termed the "Inverse Fresnel Zone plate" (i.e the n-th radius of the Fresnel zone plate multiplied by the n-th radius of the inverse Fresnel zone plate is a constant for all *n*). These two kinds of division complement each other with respect to the nature of patterns to be discriminated. The first kind of division has rings that become very narrow for high spatial frequency values thus making it a good rotation-invariant filter for patterns having their important features at high frequencies. Conversely, the second choice will be suitable for filtering information at low spatial frequencies. An intermediate case may be treated with filters having q = 1 where the width of the rings is constant. This analysis is reminiscent of the procedures utilized in ref. 16 where a specific circular harmonic was chosen for each recognition task depending on the objects to be dealt with.

The number of the rings in each filter may also be chosen in a flexible way utilizing optimization algorithms. However, to avoid unnecessary complications at this stage we made the number of rings equal to the number of patterns that makes the solution of the equations (15) unique. Other implications of this subject will be addressed in a subsequent work.

VI. Discussion

A simplified approach to optical pattern recognition was proposed to make its practical application more feasible. Emphasis was placed on rotation-invariant pattern recognition and its intrinsic limitations Some of the aspects treated have quite general implications. For example, it was shown that integral transmission detection is a poor measure for pattern distinction. Therefore the present procedure, like holographic matched filtering, relies on the intensity distribution over the output plane. Further research is required for the determination of the actual influence of various parameters such as information content and possible phase variation in the filter

The extension of the present method for class discrimination is, in principle, a straight forward procedure. For example, to implement a mask that determines whether a certain pattern belongs to a subset (A) one might superpose all the mask vectors belonging to that subset

$$M_a = M_{a1} - M_{a2} - \dots +M_{an} \tag{22}$$

Although mathematically this relation is quite simple, one should keep in mind the need for filter normalization required by the physical limitations

In a subsequent publication the subject of scale-invariant pattern recognition will be addressed using a similar approach.

It is a pleasure to thank G. Daniels for performing the photographic work involved in this investigation.

References

1. G. L. Rogers, "Noncoherent optical processing" John Wiley, NY, 1977.
2. See for example S. P. Almeida and G. Indebetouw "Pattern recogition via complex spatial filtering" in "Applications of Optical Fourier Transforms" H. Stark, Ed. Academic Press 1982, Orlando.
3. A. B. VanderLugt, "Signal detection by complex spatial filtering" IEEE Trans. Inf. Theory IT-10, 139-145, (1964).
4. H. J. Caulfield and W. T. Maloney, "Improved discrimination in optical character recognition" Appl. Opt. 8, 2354-6, (1969)
5. B. Braunecker, R. Hauch and A. W. Lohmann "Optical Character recognition based on nonredundant correlation measurements" Appl. Opt. 18, 2746-53, (1979)
6. C. F. Hester and D. Casasent, "Multivariant technique for multiscale pattern recognition" appl. Opt. 19, 1758-61, (1980)
7. H. J. Caulfield and R. Haimes, "Generalized matched filtering" Appl. Opt. 19, 181-3, (1980).
8. H. J. Caulfield, "Linear combinations of filters for character recognition: a unified treatment" Appl. Opt. 19, 3877-8, (1980).
9. S. P. Almeida and J. K. Tzong Eu "Water pollution monitoring using matched spatial filters" Appl. Opt. 15, 510-515 (1976).
10. J. C. Vienot and J. Duvernoy, "Optical computing and investigations on writing" Appl. Opt. 15, 523-529, (1976).
11. Z. H. Gu, J. R. Leger and S. H. Lee, "Optical implementation of the least-squares linear mapping techniques for image classification" JOSA 72, 787-793, (1982).
12. B. V. K. Vijaya Kumar, "Efficient approach to designing linear combination filters" Appl. Opt. 22, 1445-48, (1983).
13. F. T. S. Yu and J. Lu, "Real-time optical scanning correlator" Appl. Opt. 23, 3109-16 (1984)
14. D. A. Gregory and H. K. Liu, "Large-memory real-time multichannel multiplexed pattern correlator" Appl. Opt. 23, 4560-70, (1984).
15. S. P. Almeida and J. K. Tzong Eu, "Rotating wedge spatial filter" Appl. Opt. 17, 163, (1978).
16. B. Pernick, R. E. Kopp, J. Lisa, J. Mendelsohn, H. Stone and R. Who'ers, Screening of cervical cytological samples using coherent optical processing" Appl. Opt. 17, 21-31 (1978).
17. Y. N. Hsu, and H. H. Arsenault, "Optical pattern recognition using circular harmonic expansion" Appl. Opt. 22, 4016-19, (1982).
18. H. H. Arsenault and C. Delisle, "Contrast-invariant pattern recognition using circular harmonic components" Appl. Opt. 24, 2072-2075, (1985) and references there.
19. G. F. Schils and D. W. Sweeney, "Rotation invariant correlation filtering", JOSE 2A, 1411-1418, (1985).
20. A Furman and D. Casasent, "Bipolar incoherent optical pattern recognition by carrier encoding" Appl. Opt. 18, 660-665, (1979).

Circular harmonic phase filters for efficient rotation-invariant pattern recognition

Joseph Rosen and Joseph Shamir

A generalized approach for pattern recognition using spatial filters with reduced tolerance requirements was described in some recent publications. This approach leads to various possible implementations such as the composite matched filter, the circular harmonic matched filter, or the composite circular harmonic matched filter. The present work describes new examples leading to very high selectivity filters retaining rotation invariance and reduced requirements on device resolution. Computer simulations and laboratory experiments show the advantages of this approach.

I. Introduction

Conventional methods of optical pattern recognition suffer from the requirement of high resolution recording materials and distortion sensitivity. In some recent publications[1,2] a new, general procedure was introduced that may be employed for generating spatial filters with reduced resolution requirements. Partial and complete rotation-invariance was demonstrated in computer simulations and laboratory experiments employing bipolar amplitude filters, phase-only filters, and composite phase filters.

In this work we show that a good example of the new procedure is the circular harmonic component filter in its regular complex amplitude form and also in its phase-only form. These filters can be used as the basic constituents in a composite filter where the advantages of phase-only filters and complex amplitude filters are combined. The initial goal of our research project,[1] i.e., the use of reduced information content filters is preserved together with a high degree of distortion invariance. In this paper we demonstrate rotation invariance only but preliminary experiments indicate that scale invariance can be approached with a similar procedure.

II. Rotation-Invariant Filter Design

Our objective is to find an efficient filter, determined by a characteristic function $g(x,y)$, that can recognize a pattern $f(x,y)$ in the presence of other patterns. The recognition criterion will use the conventional correlation function

$$C(x_0,y_0) = \int f(x,y)g^*(x - x_0,y - y_0)dxdy, \qquad (1)$$

and in particular its value at the origin

$$C(0) = \int_0^\infty \int_0^{2\pi} f(r,\theta)g^*(r,\theta)rd\theta dr, \qquad (2)$$

where we converted to polar coordinates for convenience in treating the subjects of rotation and scale invariance. Defining this equation as the system response one may also define the response for an object rotated by an angle α,

$$C(0;\alpha) = \int_0^\infty \int_0^{2\pi} f(r,\theta + \alpha)g^*(r,\theta)rd\theta dr. \qquad (3)$$

Ideally one would like to keep $C(0;\alpha)$ constant regardless of the value of α. However, since this requirement is usually beyond practical limits one has to look for various compromises. For example, the performance of circular harmonic component filters has been investigated for completely rotation-invariant pattern recognition by Arsenault and Sheng.[3] A filter made for a single circular harmonic component yields a correlation

$$C(0;\alpha) = K \exp(jn\alpha), \qquad (4)$$

where K is a constant and n is the order of the harmonic. For intensity measurements this response is quite satisfactory.

In the present approach we turn around the argument and start by defining the required response,

Joseph Rosen is with Technion—Israel Institute of Technology, Haifa 32000, Israel, and J. Shamir is with University of Alabama in Huntsville, Center for Applied Optics, Huntsville, Alabama 35899.

Received 4 August 1987.
0003-6935/88/142895-05$02.00/0.

$C(0;\alpha)$. Considering this response as a function of the variable α it can be decomposed into a Fourier series:

$$C(0;\alpha) = \sum_{n=-\infty}^{\infty} c_n \exp(jn\alpha). \qquad (5)$$

Working in the Fourier plane it is useful to represent the Fourier transform of the input patterns and the characteristic filter functions in a circular harmonic decomposition:

$$F(\rho,\phi) = \sum_{n=-\infty}^{\infty} F_n(\rho) \exp(jn\phi), \qquad (6)$$

$$G(\rho,\phi) = \sum_{n=-\infty}^{\infty} G_n(\rho) \exp(jn\phi), \qquad (7)$$

where ρ and ϕ are the polar coordinates in the Fourier plane. It is easy to show that the value of the correlation function at the origin [Eq. (3)] can also be written in the simple form

$$C(0;\alpha) = \int_0^{\infty}\int_0^{2\pi} F(\rho,\phi + \alpha)G^*(\rho,\phi)\rho d\rho d\phi. \qquad (8)$$

Comparing this with Eq. (5) and using the orthogonality of the exponentials we obtain

$$\sum_{n=-\infty}^{\infty} c_n \exp(jn\alpha) = \sum_{n=-\infty}^{\infty} \int_0^{\infty} F_n(\rho)G_n^*(\rho) \exp(jn\alpha)\rho d\rho, \qquad (9)$$

or

$$c_n = \int_0^{\infty} F_n(\rho)G_n^*(\rho)\rho d\rho. \qquad (10)$$

Following the traditional way of matching a certain circular harmonic component filter to the circular harmonic component in the object one may do the same in the Fourier plane by taking $G_n(\rho) = F_n(\rho)$. This filter, however, does not take into consideration the fact that the energy content in each harmonic component is very object dependent causing an appreciable reduction in light efficiency and filter selectivity. To remedy this drawback we may introduce a weighting factor into each characteristic filter function. Also, recalling the high efficiency and selectivity obtained with phase-only filters[4,5] one is tempted to use the phase information as the major contributor for generating the filters. Thus we define the phase-only characteristic circular harmonic functions,

$$G_n(\rho) = \frac{\int_0^{2\pi} F(\rho,\phi) \exp(jn\phi)d\phi}{\left|\int_0^{2\pi} F(\rho,\phi) \exp(jn\phi)d\phi\right|}; \qquad \rho_l < \rho < \rho_h, \qquad (11)$$

where ρ_h is the size of the filter and the indistinguishable low frequency signal has been eliminated (i.e., $G_n = 0$ outside the noted region). The useful interval depends on the pattern to be recognized and should be chosen in such a way that it contains the distinguishing information.

The most convenient way to proceed is to invoke a specific example. Previous experiments with block letters indicated that it is most difficult to distinguish between the letters P and F such as shown in Fig. 1. Thus it is interesting to investigate this difficult case with various filters made to recognize one of these letters against the other. In a computer experiment filters were generated to recognize the letter P from the input pattern of Fig. 1. The performance with a regular matched filter is shown in Fig. 2(a) with the autocorrelation peak normalized to unity. It is clear that the cross correlation with F is quite high, much higher than the correlation with the rotated P. The autocorrelation peak of a phase-only matched filter is 54 times as high [Fig. 2(b)] but the cross correlation with F is high too, again much higher than that with the rotated P. A circular harmonic component with $n = 0$ produces the output pattern shown in Fig. 2(c), demonstrating complete rotation invariance but not very good selectivity. The improvement obtained by using a phase-only circular harmonic component filter of the type represented by Eq. (11) is indicated by Fig. 2(d). Low frequency suppression for the two last experiments was the same.

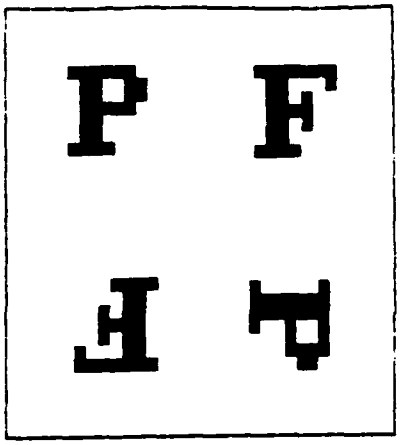


Fig. 1. Input pattern for the computer experiments from which the letter P should be recognized.

The experimental results shown in Fig. 2 are, respectively, summarized in lines 1–4 of Table I. I_P is the autocorrelation peak intensity normalized to 1 for the classical matched filter while I_P/I_F is the ratio between the peak for P to that for F. The last column indicates if the filter is completely rotation invariant or not.

III. Phase Amplitude Composite Filter Generation

The good performance of the new filter is still deteriorated by the presence of a cross-correlation peak. To suppress this peak one must also include in the filter function some information about the pattern to be rejected. This can be achieved by using the concept of the composite filter[6] as also implemented for the circular harmonic filters.[7] Figure 3 is the output pattern obtained by using such a rotation-invariant complex amplitude filter (see also line 5 in Table I). In principle one could use the same procedure with the new phase filters; however, due to the rapid fluctuations of the intensity over the output plane this is too difficult. Thus to suppress the cross-correlation peaks one may

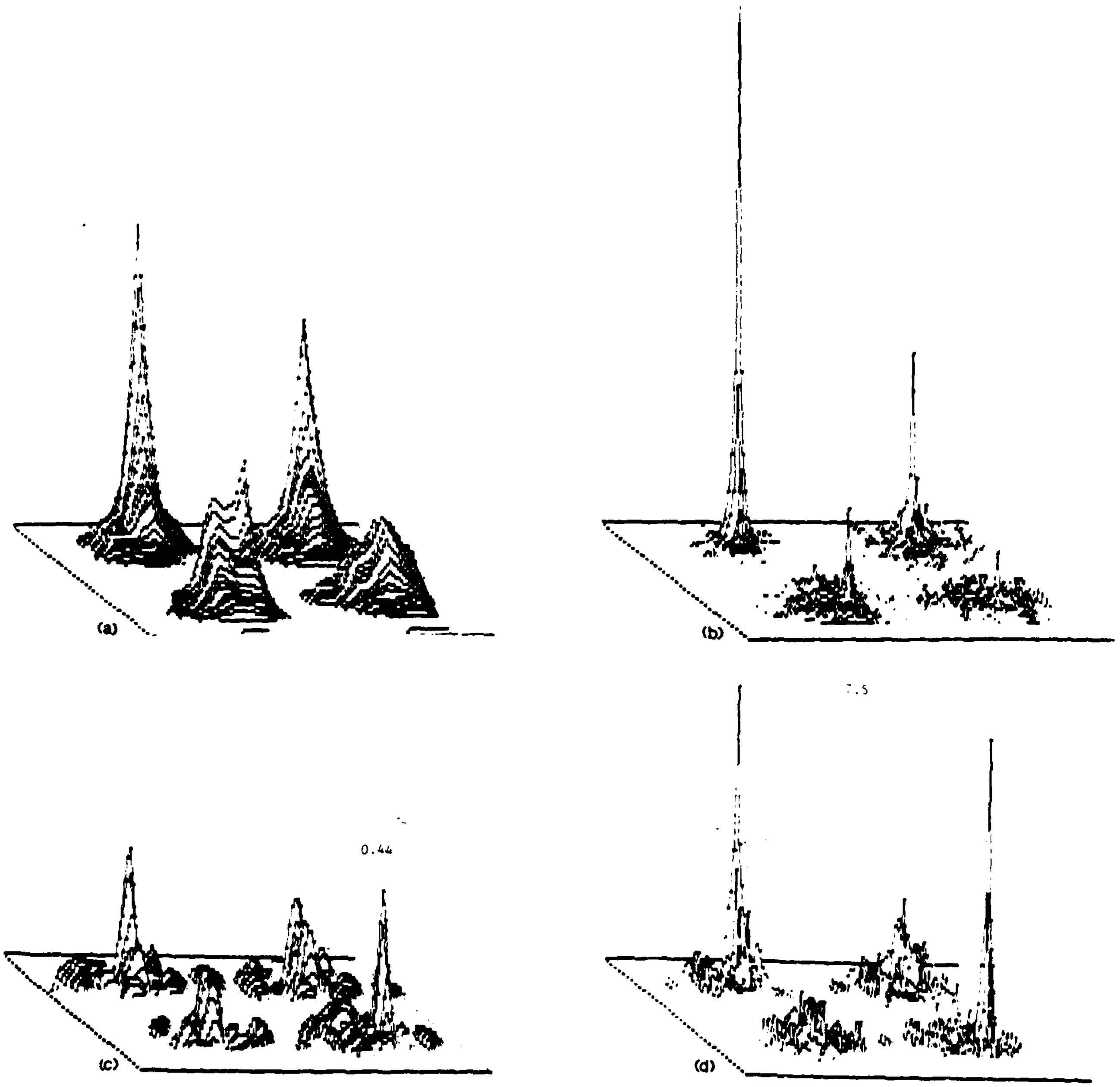


Fig. 2. Output distribution for (a) regular matched filter; (b) phase-only matched filter; (c) harmonic component ($n = 0$) filter; and (d) harmonic component ($n = 0$) phase-only filter.

Table I. Performance Comparison for the Various Filters (See Text for Details); Parameters ν_1 and ν_2 Define the Weight of Each Component of the Composite Filters

Filter	$I_P = \lvert C(0,\alpha\rvert^2$	I_P/I_F	Rotation invariant
(1) Matched filter	1	1.42	No
(2) Phase-only filter	54	2.8	No
(3) Circular harmonic component filter $N = 0$	0.44	1.7	Yes
(4) Phase-only circular harmonic component filter $N = 0$	7.5	3.5	Yes
(5) Composite filter $= \upsilon_1 F_3 + \upsilon_2 F_3$	0.55	2.0	Yes
(6) Composite filter $= \upsilon_1 F_4 + \upsilon_2 F_1$	3	5.5	Yes

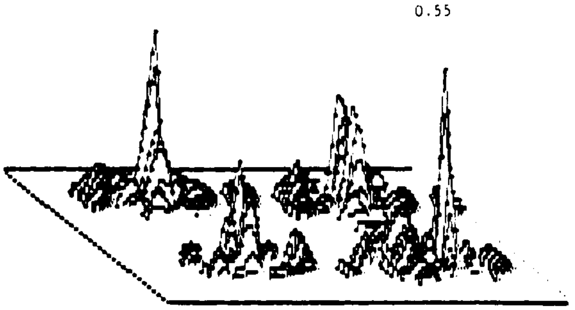


Fig. 3. Output distribution with a harmonic component composite filter.

use a smother characteristic function for the unwanted patterns (F in this case) in a composite filter. One possible choice can be the circular harmonic component filter employed in generatinr the output of Fig. 2(c). This way we may compose a filter where we utilize the high light efficiency of phase-only filters for the pattern to be recognized and modify it with the complex functions of the patterns to be rejected.

With the above considerations in mind we generate the characteristic filter function for P according to Eq. (11) for the $n = 0$ circular harmonic. For the same circular harmonic component we generate the circular harmonic filter for F according to the relation

$$G_F(\rho) = \int_0^{2\pi} F_F(\rho,\phi)d\phi, \qquad (12)$$

and combine them in a composite filter.

A scan along the diagonal of the filter is shown in Fig. 4. It turns out that for real objects, as is the situation in our experiments, the zero-order phase-only circular harmonic has only the values zero or π leading to a binary, bipolar amplitude filter with values 1 and -1. The plot in Fig. 4 represents such a filter made for P, modified by the complex filter function prepared for the zero-order circular harmonic of the letter F. The output pattern for this filter is shown in Fig. 5.

The measurements performed on the outputs of Fig. 5 are summarized in line 6 of Table I. While line 5 represents the results for a filter composed of two characteristic functions that served as filters in line 3, the filter for line 6 is made out of a P function corresponding to the filter in line 4 combined with an F function corresponding to a filter of line 3. The improved discrimination characteristic of the new composite filter compared to Figs. 3 and 2(d) is evident.

IV. Laboratory Experiments

To verify the practicability of the new procedure the computer experiments were repeated in the laboratory. We employed the same IBM PC that was used in the simulations to generate the input pattern of Fig. 6(a) and holographic filter functions like the one shown in Fig. 6(b). To generate the filters the Fourier plane was sampled into 64 rings of equai width and the holograms were plotted on a regular dot printer. The working patterns were obtained by a 25-fold photographic reduction onto a regular photograp^hic film.

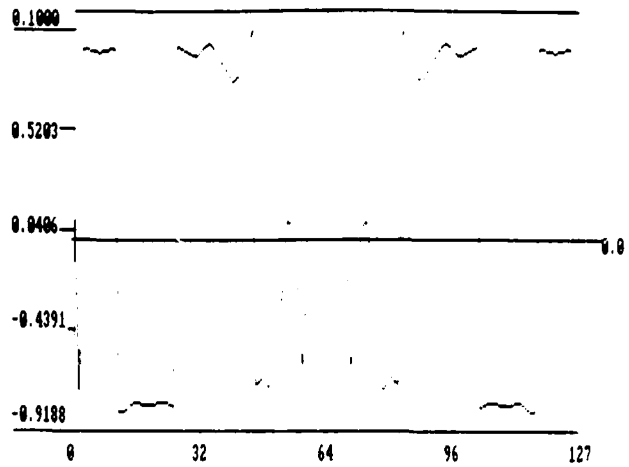


Fig. 4. Bipolar amplitude scan along one diameter of a phase amplitude harmonic component filter.

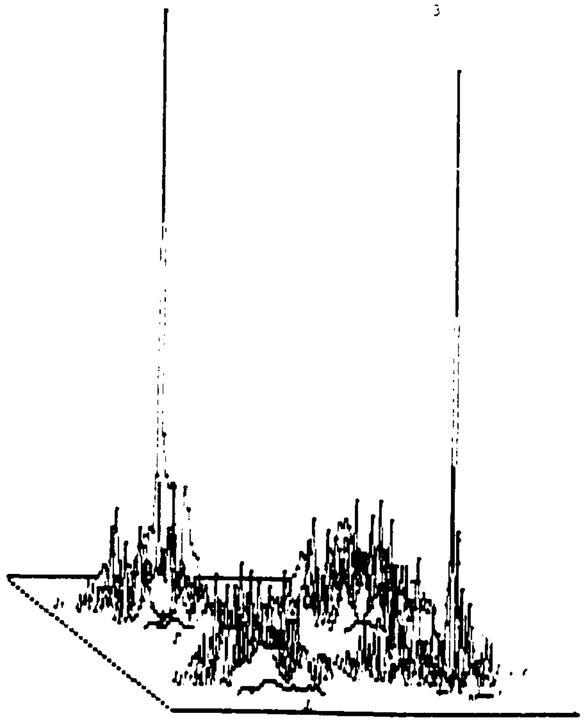


Fig. 5. Output pattern for the filter of Fig. 4.

Figure 6(c) shows the output pattern for a phase-only circular harmonic component filter (corresponding to line 4 in Table I) superposed by a line along which the intensity scan of Fig. 6(d) was obtained. Figures 6(e) and (f) are the corresponding patterns for the composite filter of Fig. 4 (line 6 in the table). The correspondence with the computer calculations is excellent. Note that the correlation peaks appear at the centroids of the recognized patterns that are shifted during rotation. It is interesting that the cross correlation with the additional letter O was also reduced with the composite filter.

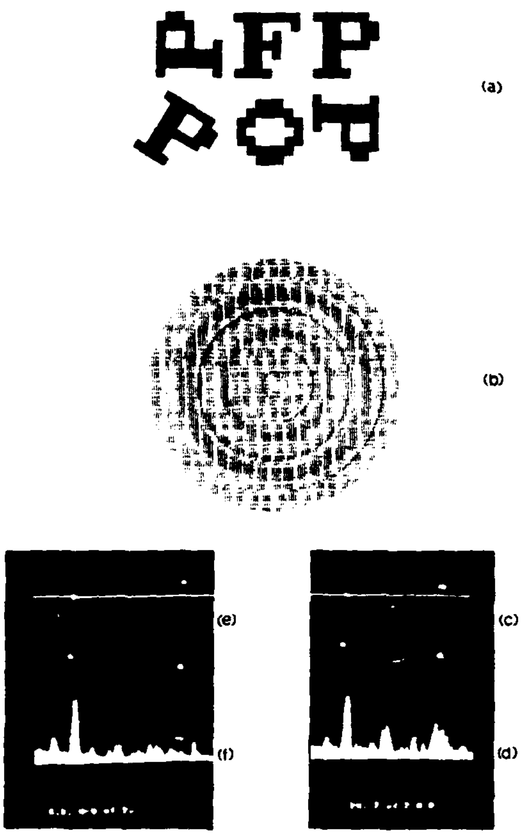


Fig. 6. (a) Input pattern for laboratory experiment; (b) filter made for recognizing *P*; (c) output pattern with phase-only harmonic component filter superposed by a line along which the intensity scan of (d) was taken; (e) output for a phase-amplitude harmonic component composite filter of Fig. 4 with intensity scan (f).

V. Conclusions

In this work we introduced a new kind of phase-only filter, the phase-only circular harmonic component filter and the circular harmonic component phase amplitude composite filter. The selectivity and light efficiency of the composite filters were improved by combining the advantages of the phase-only filters with those of the complex amplitude filters. The superior performance of these filters was demonstrated by computer simulations and laboratory experiments. We worked with the zero-order harmonic because the letter *P* had a very large fraction of its energy in this harmonic. For the detection of *F*, for example, a higher harmonic is better. In any case, a set of filters for a specific job may include many harmonic orders. However, to preserve rotation invariance, each filter should contain information using the same harmonic component of all the input patterns. The experiments described in this paper are only a sample of those actually performed and they represent the most problematic cases.

The initial goal of the present research project of employing low resolution devices was preserved and demonstrated by using a simple dot printer for the generation of the filters and regular photographic film in the actual experiments. It is also worthwhile noting that this entire paper represents just a new example of the general procedure outlined in Ref. 1.

This work was partially supported by contract N00014-86-K-0591 with the Office of Naval Research.

References

1. J. Shamir, H. J. Caulfield, and J. Rosen, "Pattern Recognition Using Reduced Information Content Filters," Appl. Opt. 26, 2311 (1987).
2. J. Rosen and J. Shamir, "Distortion Invariant Pattern Recognition with Phase Filters," Appl. Opt. 26, 2315 (1987).
3. H. H. Arsenault and Y. Sheng, "Properties of the Circular Harmonic Expansion for Rotation-Invariant Pattern Recognition," Appl. Opt. 25, 3225 (1986), and references therein.
4. J. L. Horner and P. D. Gianino, "Applying the Phase-Only Filter Concept to the Synthetic Discriminant Function Correlation Filter," Appl. Opt. 24, 851 (1985).
5. J. L. Horner and J. R. Leger, "Pattern Recognition with Binary Phase-Only Filters," Appl. Opt. 24, 609 (1985).
6. H. J. Caulfield and W. T. Maloney, "Improved Discrimination in Optical Character Recognition," Appl. Opt. 8, 2354 (1969).
7. G. F. Schils and D. W. Sweeney, "Iterative Technique for the Synthesis of Distortion-Invariant Optical Correlation Filters," Opt. Lett. 12, 307 (1987).

Weight discretization in backward error propagation neural networks[†]

E. Fiesler, A. Choudry, and H.J. Caulfield

Center for Applied Optics / Department of Computer Science
University of Alabama in Huntsville (U.A.H.)
Huntsville, Alabama 35899, U.S.A.

Abstract

The number of interconnections in a (fully connected) backward error propagation neural network grows quadratically with the number of neurons in the network. The memory (and time) requirements for handling a large number of interconnections can therefore become a serious impediment for simulations and ımplementations of neural networks. Another problem is that the media used by most neural network implementations (neural computers) have only a limited ability to discriminate intensity levels. In order to represent neural networks efficiently in optical implementations (optical computers) and analog electronic implementations, the set of possible values an interconnection strength (weight) can have, should be small. To abate these problems, the possibility of discretizing the weights of neural networks is investigated. Weight discretization will impair the performance of the neural network. This can be compensated by increasing the number of neurons and/or the number of hidden layers. A new discretization method is developed and its performance is compared to others.

1 Introduction

1.1 Background and definitions

Perceptron like neural networks can be trained by teaching them patterns. A pattern consists of a set of elements (*pixels*). Each pixel can assume a continuous or a discrete value. In the discrete case[1], the possible set of pixel values is often limited. Typical pixel value sets, used in artificial neural networks, are: $\{0,1\}$ and $\{-1,1\}$. Usually, a pattern is presented to the neural network by feeding each of the pixels to a different input neuron, i.e. a neuron of the first layer of the neural network. Therefore the number of input neurons is equal to the number of pixels in the pattern.

Most artificial neural network simulations consist of two phases : a *training* or *learning phase*, and a *recall* or *use phase*. During the training phase patterns are presented to the network. The interconnection strengths (also called synaptic strengths or weights) of the neural network are adapted conform these patterns by means of a neural network learning rule (as for example the backward error propagation learning rule). If the weights are stabilized, the network is called *fully trained*. During the recall phase input patterns are presented to the neural network. Based on the fixed weights, corresponding outputs, which are the activation values of the neurons of the highest layer, are generated by the network. This form of neural network training is called *off-line training*. Off-line training is often crucial, since it separates the normally time consuming training from the recall process and therefore speeds up the use of the neural network tremendously.

In 'neural network learning rules with a teacher', two patterns are presented to the network: an input pattern and a *target pattern* (the 'teacher') which is the desired output for the neural network. In these networks the total output of the network has to converge towards the target pattern; i.e. the activation values of the output neurons have to converge towards the pixel values of the target pattern. In *auto-associative learning* the input patterns are the same as the target patterns. Auto-associative learning is therefore used to train the neural network to remember a set of patterns. One of the main applications of auto-associative learning is image reconstruction or recalling a pattern if only a partial or disabled input is available. For example: a knowledge base which can handle fuzzy data.

In *hetero-associative learning* the input and target patterns are usually different. Hetero-associative learning is therefore used to train the network to associate each of the input patterns with its corresponding target pattern. For example: association of geological information of a certain geographical area with the presence of fossil fuels there.

[1] Be aware of the difference between discrete weights and discrete pixel values.

1.2 Problem definition

Since the number of interconnections in a fully interconnected back-propagation neural network grows quadratically with the number of neurons in the network, the storage (and time) needed for handling them is often a problem for neural network simulations and implementations. Discrete values (from a limited set) use less storage and can be handled more economically than continuous values. Therefore, discretization of the weights is investigated.

Discretization is essential for all kinds of implementations of neural networks, since most information media used can only discriminate a small set of intensity levels. For example optical implementations (optical computing) [Caulfield-88] and analog electronic implementations [Thakoor-86].

The research goals for the work presented in this paper are to develop discretization methods for back-propagation neural networks, to create a software environment for the simulation of neural network weight discretization, to test the discretization methods by computer simulation, and to find rules of thumb for expanding the neural network in order to compensate for the loss of information capacity due to the discretization of the weights. Before going to the discretization methods, some related work is discussed.

1.3 Prior work

1.3.1 Hopfield model

In his famous 1982 paper, Hopfield [Hopfield-82] studied a 'clipped' weight matrix (T_{ij}). He replaced T_{ij} by ±1, the algebraic sign of T_{ij}. The purposes were to examine the necessity of a linear synapse supposition (by making a highly nonlinear one) and to examine the efficiency of storage. He found little performance[2] degradation. The number of recallable patterns was (analytically) $\frac{2}{\pi}$ of the number with linear T_{ij}'s. Thus severe discretization causes only mild degradation. To restore performance, the number of neurons would have to be increased by $\frac{\pi}{2}$.

[2] In this paper, the 'performance' of a neural network is its ability to learn (and recall) a certain amount of information.

1.3.2 Winner-take-all-models

Stirk et al. [Stirk-87] addressed a variety of non-Hopfield models from the viewpoint of performance sensitivity to analog optical inaccuracies. The results for simple winner-take-all networks are bad. Furthermore "big N" cases ($N = 64$) are significantly worse than "small N" ($N = 16$) cases. Optics seems advantageous over electronics only for very large N, say, 10^4 to 10^6. This means optics is accurate enough only for small N, but small N is probably better done electronically.

1.3.3 Farhat's adaptive method

In a paper showing how to implement large neural networks ($10^3 - 10^4$ neurons), Farhat et al. [Farhat-86] reformulate the Hopfield model for two-dimensional inputs and outputs and four-dimensional interconnects. They clip the interconnection matrix in various ways $\{0,1\},\{-1,1\},\{-1,0,1\}$, etc. and find that with "adaptive thresholds" the $\{0,1\}$ interconnection pattern (easy to implement optically and electronically) can achieve the same performance level as a multivalued interconnection pattern. In effect, they have restored the $\frac{2}{\pi}$ loss observed by Hopfield by using adaptive thresholds.

1.3.4 Summary of prior work

What is known from these prior studies is that some neural network designs are far more noise prone than others and that compensatory methods such as adding more neurons or allowing adaptive thresholds can restore the performance of the network.

2 Discretization of back-propagation networks

2.1 Approach

Among the multi-layer neural network learning rules capable of both auto-associative, and hetero-associative learning, the backward error propagation learning rule, also known as back-propagation or error propagation

[Werbos-74] [Rumelhart-86], is the most widely used and is simple to use [Hecht-Nielsen-88]. The back-propagation learning rule was therefore chosen for the experiments. The experiments are based on the discretization methods which are described in paragraph 2.2.

The number of neurons per layer can vary; N_l indicates the number of neurons in layer l ($1 \leq l \leq L$), where L is the total number of layers (or slabs) in the network including the input and the output layer. The interconnection weights between two layers of a neural network can be represented by a matrix $W_{l,ij}$, here l represents the level of the matrix ($1 \leq l < L$). The level l is the ordinal number of the lower one of the two layers connected by $W_{l,ij}$. The indices i and j determine the ordinal number of the neuron in the lower and upper layer respectively. The weights ($W^c_{l,ij}$), as used in ordinary back-propagation models, can theoretically assume any (continuous) value:

$$-\infty < W^c_{l,ij} < \infty.$$

The used discretization methods produce discrete weights ($W^d_{l,ij}$). In general there are D discretization levels, where D is a finite integral number. Since the set of desired discretization values can be mapped on any sequence of numbers using a bijection, any set with the same cardinal will satisfy. In this paper the choice is made for a sequence of consecutive integers equally divided among positive and negative numbers :

$$W^d_{l,ij} \in \left\{ n - \left\lfloor \frac{D+1}{2} \right\rfloor \;\middle|\; n = 1, 2, ..., D \right\}.$$

Discretization of the weights will impair the performance of the neural network, because there is a loss of information capacity. This is compensated by increasing the number of neurons and/or the number of hidden layers of the network. In other words: discretized weights contain less information than continuous ones; this is compensated by using more of them. The used discretization methods are discussed in the next paragraph.

2.2 Three discretization methods

2.2.1 The multiple-thresholding method

The multiple-thresholding method is the simplest of the three discretization methods used. It starts by fully training the neural network, using the back-propagation learning rule; i.e. iterate (over steps 2 till 4 of the algorithm in appendix A) until *convergence*[3] is reached. Then discretize the continuous weights into discrete valued weights using a nonlinear function (usually a multiple-threshold). The weight matrices so obtained are referred to as the *discrete network*. The original set of weight matrices with continuous weights is called the *continuous network*. Chiueh and Goodman [Chiueh-88] have applied this method using three discretization levels. They found that about 15%-50% of the networks did not work.

2.2.2 The direct discretization method

In the direct discretization method, the neural network is initialized with discrete weights, which have random values within the discrete range. The forward propagation is similar to the normal back-propagation learning rule (step 2 of the algorithm in appendix A). During the backward propagation (step 3 of the algorithm in appendix A), the weights are updated only if the difference in weight ($\Delta W^d_{l,ij}$) is big enough to change the weight into one of the other possible discrete values. This method does not work for the standard back-propagation learning rule (see appendix B for a proof).

2.2.3 The continuous-discrete learning method (CDLM)

This new developed method, schematically shown in figure 1, starts off with the multiple-thresholding method (paragraph 2.2.1, and (a) to (f) in figure 1). Next the original input pattern (a) is fed (h) into the discrete network (g). The outputs obtained by forward propagation (step 2b of appendix A) are compared (i) with the target pattern (e) and the errors (δ's) are back-propagated (j) through the **continuous network** (c). Next, the weights of the continuous network are discretized (f) as before and the process starts all over again until the system reaches convergence. The fully trained discrete network (g) can now be used for the recall phase.

This approach leads to an increase in the total number of iterations

[3] From now on "reaching convergence" will stand for reaching of the convergence criterion (see paragraphs 1.1 and 3.1) or another limiting (normally time-based) factor, e.g. the maximum number of iterations allowed. "Convergence" itself will stand for converging into the desired range (ϵ-range), as opposed to convergence to any value.

needed. The process can be speeded up by skipping the full training of the continuous network, since starting with a fully trained continuous network is convenient, but not necessary.

3 Evaluation of the discretization methods

3.1 The back-propagation model used

The experiments performed are based on the back-propagation learning rule. The characteristics of the back-propagation model used are: it is fully connected between adjacent layers, it has no *intralayer connections* i.e. connections between neurons in the same layer, and no *supralayer connections* i.e. connections between neurons that are not in adjacent layers [Rumelhart-86, figure 8.3]. The following assumptions are made: a negative weight is inhibitory, a zero weight means no connection and a positive weight is excitatory. Thus, in spite of the fully connectedness (between adjacent layers), the situation of two neurons without an interconnection can be represented theoretically in this way.

The patterns used to train the network were free of noise. They are presented to the neural network as a set of pairs of patterns. Each pair consists of an input pattern and its corresponding target output pattern. A pattern consists of a rectangular matrix of pixel values (height × width) which is mapped onto the one dimensional set of input neurons (the neurons in layer one). Let h^i be the height of the input patterns and w^i the width (i stands for input). The pixel value of input pattern j ($p_{j,mn}$) is mapped on input neuron $(m-1)w^i + n$, where m indicates the row of the pixel in the pattern ($1 \leq m \leq h^i$) and n the column ($1 \leq n \leq w^i$). The patterns in the set, which are presented in the order they are provided by the user, are fed repeatedly into the input neurons of the neural network until convergence is reached. The convergence criterion used is: when all the activation values of the output neurons reach their ϵ-range. An ϵ-range is the range near a desired output, determined by the *deviation parameter* (ϵ). The deviation parameter is the maximum amount that an output activation value may deviate from the target pattern value.

3.2 Implementation

3.2.1 Software specification

In order to perform the discretization experiments with all the necessary flexibility, a portable (machine independent) back-propagation software environment was developed by the author using the PASCAL programming language [Jensen-78]. The main part of the software has been developed on a personal computer. When some of the experiments took more than 24 hours to run on the personal computer, changing over to a Cray X-MP/24 supercomputer seemed a good idea.

Some of the flexibility criteria for the software environment were : the capability of handling both auto-associative and hetero-associative learning, changing the pattern size (height and width), the number of patterns, the learning rate (η), the number of layers, the number of neurons per layer (for each hidden layer), the number of discretization levels, the deviation parameter, and the maximum number of iterations for the (continuous-discrete) learning method, also the ability of choosing a discretization method, an initialization scheme for the weights, and a pixel value set.

The most important outputs of the simulation system (for both the continuous and the discrete network) are:

- the stop criterion : whether the desired output is reached (within the ϵ-range) or the number of iterations reached its maximum
- the output values (activation values of the output neurons) after each iteration, if desired
- the number of iterations made
- the number of errors made (output activation values that reached undesired values)
- the number of output neurons that did not reach the desired output (within the ϵ-range)
- the maximum deviation (between actual output activation value and the desired output value)

The user can choose which outputs are desired for specific experiments.

3.2.2 Methodology

The most promising discretization method is the CDLM, because the direct discretization method does not work, and the CDLM easily outperforms the multiple-thresholding method because the first includes and improves the second method.

Two approaches were taken to test both the CDLM and the multi-threshold method. First a systematical 'search' through the state space of possible experiments. The starting position was the smallest network possible and using two discretization levels, since this is the preferred number for most neural network implementations. The next variable to vary is therefore the pattern size which is the same as the input size. Then both auto- and hetero-associative learning were tested. The number of patterns was the next variable to vary. This meant starting off with a two layer system, which would be enlarged in further experiments. The number of possible experiments was growing exponentially, a second approach was taken.

Here, the collection of patterns was fixed. This means a fixed number of patterns, a fixed pattern size, a fixed number of input and output neurons and, in this case, a choice was made for hetero-associative learning. The central parameter in this approach is the number of discretization levels.

3.2.3 Parameter definition

This paragraph discusses the parameters which were kept constant in most of the experiments. For perfect recall (i.e. output activation values are within their ϵ-range), using noise free inputs, it turned out that the higher the learning rate the faster the convergence. Besides dedicated experiments, the value of the learning rate (η) was kept at 0.5 [Caudill-88]. The value used for the deviation parameter (ϵ) is 0.05. Random values in the range $[-0.1, 0.1]$ were used to initialize the weights. The pixel value set used is $\{-1, 1\}$. For the nonlinear function required in both the multiple-thresholding method and the CDLM, a multiple threshold with rounding off to the nearest pixel value was used. A typical figure for the maximum number of iterations is 20,000. The local thresholds (Θ's) or biases [Rumelhart-86, p331-] were kept zero.

In the second approach a number of variables were fixed in order to limit the state space. These experiments used hetero-associative learning, three by five pixel patterns, a learning rate of 0.5, and two to four layers.

3.3 Results

"Being able to learn and perfectly recall (associate) a set of noise free patterns" is taken as a measure for comparing the performances of continuous and discrete networks.

What could be expected intuitively, is confirmed by the experiments: the performance of the CDLM is better than the performance of the multiple-threshold method. The outputs of the multiple-thresholding method were often outside the ϵ-range. Sometimes, wrong results were obtained when rounding[4] was applied to the outputs. In general rounding can be used to obtain a $\{0, 1\}$-result from an output neuron that did not converge into the ϵ-range.

The CDLM on the other hand usually achieves much better results. The first approach (see paragraph 3.2.2) emphasized the performance restoration of the neural network using the minimum number of discretization levels, which is two. In the case of associating one set of two patterns by the simplest network of one input neuron, one output neuron and a variable number of hidden layers and neurons in them, a two level discretization works very well (see figure 2). The two layer network does not converge to a value within the ϵ-range but gives the right answer after rounding. The performance of a three layer system with one neuron in the hidden layer is worse than the two-layered network. But adding neurons to the hidden layer increases the performance. With five neurons in the hidden layer the ϵ-range of 0.05 is reached. In figure 3 the situation for four layers is depicted. In order to reach ϵ-accuracy, the minimum number of neurons needed in the hidden layers is (5&3), (2&4), and (1&5) neurons in the second & third layer. If two patterns are stored, the graph (see figure 4) is less smooth but the same behavior can be observed. Fourteen hidden neurons are needed in the second layer to reach ϵ-accuracy. Note that for the three layered network with one or two neurons in the hidden layer faulty results are produced when rounding is used. In order to reach the right outputs after applying rounding in the case of four layers, the minimal number of units needed in the hidden layers is (2&9), (3&6), (4&5), (5&4), (6&4), (7&4), (8&3), and (9&4) neurons in the second & third layer. However, rounding gives sometimes wrong results for some hidden layer sizes larger than these minima. The number of extra neurons needed to restore the performance is relatively high. Other results showed that this relative overhead became smaller for bigger networks.

[4] From now on, "rounding" denotes rounding off to the nearest pixel value.

In some of the smaller networks the activation values of the output neurons remained constant during the discrete training. In these cases the performances of the multiple-thresholding method and the CDLM are equal.

The emphasize of the second approach is on comparing results using different numbers of discretization levels. The pattern set consists of nine pairs of character-like patterns. The continuous network could perfectly associate them all after 372 iterations.

# discr. levels	# iterations	# non-converg.	max. abberation
2	10000	67	0.88
3	10000	18	0.50
5	10000	2	0.12
7	69	0	0.00
9	47	0	0.00

This table shows that if the number of discretization levels increases, the number of non-converging outputs decreased. A perfect recall was obtained at seven discretization levels. Further increase leads to a decrease in the number of iterations needed for a perfect recall. This observation can also be made for the continuous network: adding neurons to the network leads to a faster convergence for the continuous network (less iterations needed).

Sometimes the performance of the network reached a maximum, without reaching total convergence. In order to compensate for this, the observed maximum performance is stored and used as final result.

If the CDLM starts with a full training of the continuous network, the number of iterations needed for training the discrete network varies from one to a number of iterations comparable to the number of iterations needed for the training of the continuous network. In this case, the total number of iterations needed for the CDLM is therefore one to two times that of the continuous network.

In general: addition of a new layer to the network, without increasing the total number of neurons in the network, results in a performance degradation. This can be explained by looking at the total number of interconnections before and after the addition of a new layer. If a fully connected two-layered network contains N_1 neurons in its input layer, and N_L in its output-layer, the total number of weights is $N_1 \cdot N_L$. The number of neurons needed in an additional (=hidden) layer to have the same number of connections is $\frac{N_1 \cdot N_L}{N_1 + N_L}$.

4 Conclusions and future work

Of the three discretization methods proposed, the CDLM works better than the multiple-thresholding method, and the direct discretization method is unusable. A portable neural network software environment has been created for performing the discretization experiments. As intuitively expected, the lower the discretization (more discretization levels), the better the performance of the neural network. But, using two discretization levels, as desired by optical and electronic implementations, give reasonable results. The results of a two layer neural network are usually good enough when using the CDLM.

If the CDLM starts with a fully trained continuous network, the number of iterations will be one tc [illegible] times that of the number needed for the full training of the continuous network.

Future work will consist of the search for other discretization methods. Also doing more experiments, which might bring better rules of thumb for restoring the performance of the neural network. Furthermore analytical analyzing of the discretization methods will be explored. Another goal is speeding up the simulation software by optimization and vectorization of the program code. Finally, since most of the data is multidimensional, data visualization techniques are being designed for representing the data.

Acknowledgements

The authors like to thank Drs. Ing. V. Harrand and Dr. P. Ryan and the members of the local neural network group for their helpful critique and fruitful discussions on this matter. They are also grateful to visiting professor Dr. Ir. J. Smith for his critical remarks and Drs. J. van der Zijp for proofreading this paper.

† This work is partially supported by TBE (agreement dated 10/21/86) and ONR (contract # N00014-86-K-0591).

APPENDIX A

Back(ward Error) Propagation; the formulas

In this appendix, $a_{l,i}$ represents the activation value of the neuron i in layer l of the neural network, and t_j is the target pattern value which corresponds to neuron j in the output layer.

Back-propagation consists of the following steps :

(1) Initialize the weights ($W^c_{l,ij}$) and offsets ($\Theta_{l,j}$).

(2a) $a_{l,i} :=$ input to the i-th neuron of the input layer.

(2b) Forward propagation :

$$a_{l,j} := \frac{1}{1 + e^{-(\sum_{i=1}^{N_{l-1}} W^c_{l-1,ij}\, a_{l-1,i} - \Theta_{l-1,j})}} \qquad \text{for } 2 \leq l \leq L$$

(3) Backward propagation :

$$\Delta W^c_{l,ji} := \eta\, \delta_{l+1,j}\, a_{l,i}$$

where

$$\delta_{l,j} := \begin{cases} (t_j - a_{l,j})a_{l,j}(1 - a_{l,j}) & \text{if } l = L - 1 \\ a_{l+1,j}(1 - a_{l+1,j})(\sum_{k=1}^{N_{l+2}} \delta_{l+2,k} W^c_{l+1,kj}) & \text{if } 1 < l < L - 1 \end{cases}$$

next, add $\Delta W^c_{l,ji}$ to $W^c_{l,ji}$

(4) IF no(t enough) convergence THEN GOTO (2a)

APPENDIX B

Proof of 'direct discretization method' limitations

The weight update formula for the highest level interconnection matrix (see step 3 of appendix A) is equivalent to:

$$\left|\Delta W^d_{L-1,ji}\right| = |(t_j - a_{L-1,j})a_{L-1,j}(1 - a_{L-1,j})a_{L,i}|$$

since $0 \leq a_{L,i} \leq 1$ and $0 < \eta < 1$:

$$\left|\Delta W^d_{L-1,ji}\right| < |(t_j - a_{L-1,j})a_{L-1,j}(1 - a_{L-1,j})|$$

Partial differentiation of this function shows that it has no local extremum in the open interval $(0,1) \times (0,1)$ of the plane spanned by $a_{L-1,j} \times t_j$. The maximum of this function will therefore be a boundary extremum of this interval. Since the function equals zero for both $a_{L-1,j} = 0$ and $a_{L-1,j} = 1$, two cases are left to be examined:

case 1: $t_j = 0$

$$\left|\Delta W^d_{L-1,ji}\right| < \left|-a^2_{L-1,j}(1 - a_{L-1,j})\right| = a^2_{L-1,j}(1 - a_{L-1,j})$$

$$\frac{d\left|\Delta W^d_{L-1,ji}\right|}{da_{L-1,j}} = a_{L-1,j}(2 - 3a_{L-1,j})$$

$$\frac{d\left|\Delta W^d_{L-1,ji}\right|}{da_{L-1,j}} = 0 \rightarrow \begin{cases} a_{L-1,j} = 0 & \text{minimum} \\ a_{L-1,j} = \frac{2}{3} & \text{maximum} \end{cases}$$

maximum for $a_{L-1,j} = \frac{2}{3} \quad \Rightarrow \quad \left|\Delta W^d_{L-1,ji}\right|^{max} < \frac{4}{27}$

case 2: $t_j = 1$

$$\left|\Delta W^d_{L-1,ji}\right| < \left|a_{L-1,j}(1-a_{L-1,j})^2\right| = a_{L-1,j}(1-a_{L-1,j})^2$$

$$\frac{d\left|\Delta W^d_{L-1,ji}\right|}{da_{L-1,j}} = (3a_{L-1,j}-1)(a_{L-1,j}-1)$$

$$\frac{d\left|\Delta W^d_{L-1,ji}\right|}{da_{L-1,j}} = 0 \rightarrow \begin{cases} a_{L-1,j} = \frac{1}{3} & \text{maximum} \\ a_{L-1,j} = 1 & \text{minimum} \end{cases}$$

$$\text{maximum for } a_{L-1,j} = \tfrac{1}{3} \quad \Rightarrow \quad \left|\Delta W^d_{L-1,ji}\right|^{max} < \tfrac{4}{27}$$

Both cases give the same result :

$$\left|\Delta W^d_{L-1,ji}\right| < \tfrac{4}{27} = 0.\overline{148}$$

This is the maximum weight increase possible. So, when using less than $\lceil\frac{27}{8}\rceil = 4$ discretization levels, the weights will never be updated. Since the average weight update is much smaller than this maximum, the direct discretization method is not usable for standard back-propagation neural networks.

References

[Caudill-88] Maureen Caudill,

Neural Networks Primer , part II,

AI Expert, volume 3, number 2, pages 55-61, February 1988.

[Caulfield-88] H. John Caulfield,

Vistas in Optical Neurocomputing Design ,

plenary session talk at the IEEE second international conference on neural networks,

at: San Diego, California, July 23-27, 1988.

[Chiueh-88] T.D. Chiueh and R.M. Goodman,

Learning algorithms for neural networks with ternary weights ,

will be presented at the First Annual Meeting of the International Neural Network Society,

at: Boston, Massachusetts, September 5-10, 1988.

[Farhat-86] N.H. Farhat, S. Miyahara and, K.S. Lee,

Optical analog of two-dimensional neural networks and their application in recognition of radar targets ,

from: Neural networks for computing,

American Institute of Physics (A.I.P.) conference proceedings 151, pages 146-152,

editor: John S. Denker,

at: Snowbird, Utah, April 13-16, 1986,

American Institute of Physics, New York, New York, 1986,

ISBN: 0-88318-351-X.

[Hecht-Nielsen-88] Robert Hecht-Nielsen,

Neurocomputing: picking the human brain ,

IEEE Spectrum, volume 25, number 3, pages 36-41, March 1988.

[Hopfield-82] John J. Hopfield,

Neural networks and physical systems with emergent collective computational abilities ,

Proceedings of the National Academy of Sciences of the U.S.A., volume 79, pages 2554-2558, April 1982.

[Jensen-78] Kathleen Jensen and Niklaus Wirth,

PASCAL User Manual and Report ,

Springer Verlag, New York, New York, 1978 (corrected printing),

ISBN: 0-387-90144-2 or 3-540-90144-2 (Berlin, Germany).

[Rumelhart-86] David E. Rumelhart, James L. McClelland, and the PDP Research Group,

Parallel Distributed Processing, Explorations in the Microstructure of Cognition , Volume 1: Foundations,

The MIT Press, Cambridge, Massachusetts, ("A Bradford Book"), 1986,

ISBN: 0-262-18120-7 or 0-262-18123-1 (set of two volumes).

[Stirk-87] Charles W. Stirk, S.M. Rovnyak and R.A. Athale,
Effects of noise on an Optical Implementation of an Additive Lateral Inhibitory Network ,
from: Proceedings of the IEEE first international conference on neural networks, volume III, pages 615-623,
editors: Maureen Caudill and Charles Butler,
at: San Diego, California, June 20-24, 1987,
SOS Printing, San Diego, California, 1987,
IEEE Catalog Number 87TH0191-7.

[Thakoor-86] A.P. Thakoor, J.L. Lamb, A. Moopenn, and John Lambe,
Binary synaptic connections based on memory switching in a-Si:H ,
from: Neural networks for computing,
American Institute of Physics (A.I.P.) conference proceedings 151, pages 426-431,
editor: John S. Denker,
at: Snowbird, Utah, April 13-16, 1986,
American Institute of Physics, New York, New York, 1986,
ISBN: 0-88318-351-X.

[Werbos-74] Paul Werbos,
Beyond regression: New tools for prediction and analysis in the behavioral sciences ,
Ph.D. dissertation, Harvard University, 1974.

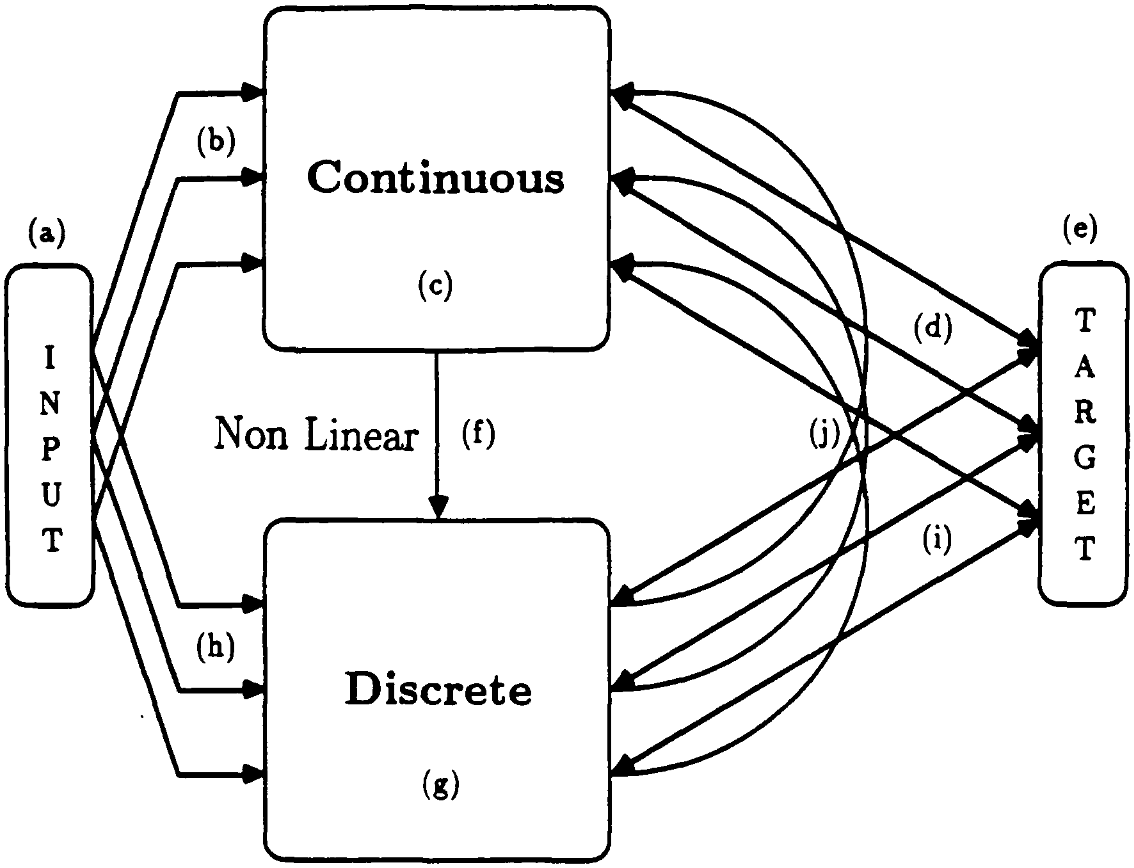


figure 1 The continuous-discrete learning method

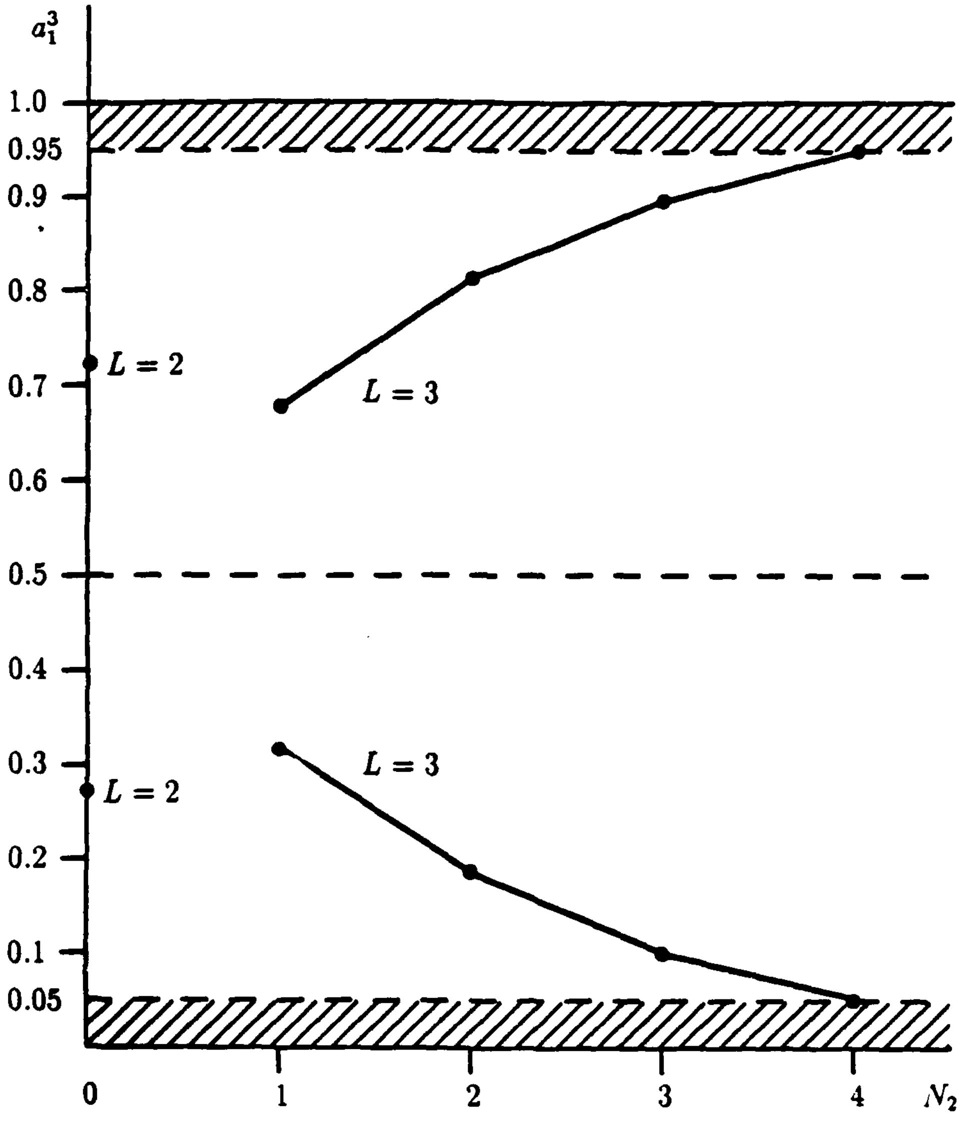


figure 2

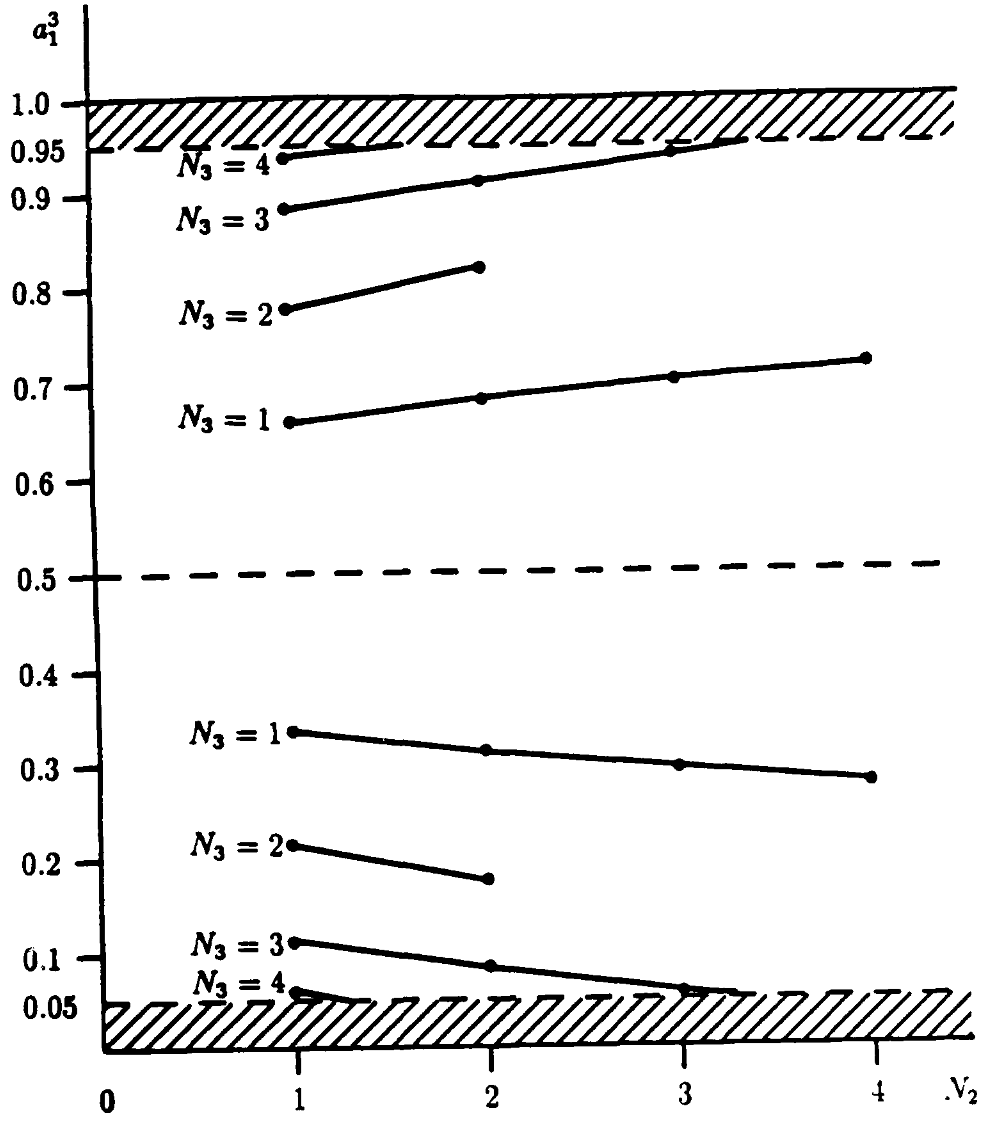

a_1^3
1.0
0.95
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.05
0
1
2
3
4
N_2
$N_3 = 4$
$N_3 = 3$
$N_3 = 2$
$N_3 = 1$
$N_3 = 1$
$N_3 = 2$
$N_3 = 3$
$N_3 = 4$

figure 3

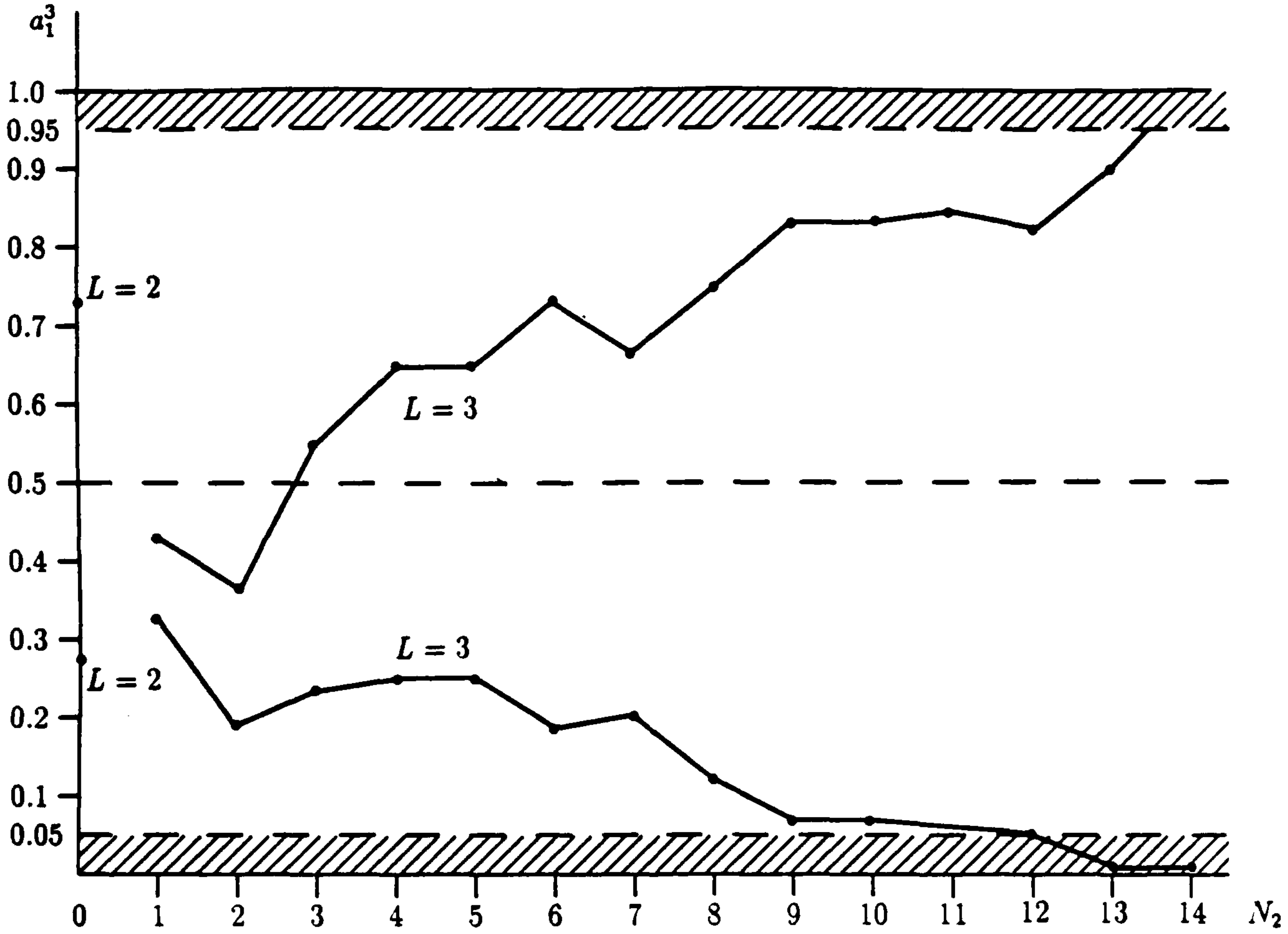

a_1^3
1.0
0.95
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.05
$L = 2$
$L = 3$
$L = 3$
$L = 2$
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
N_2

figure 4

SOME THEORETICAL UPPERBOUNDS ON THE CAPACITY OF NEURAL NETWORKS[†]

E. Fiesler[1,2], H.J. Caulfield[2], and A. Choudry[3]

[1]Department of Computer Science / [2]Center for Applied Optics /
[3]Department of Electrical and Computer Engineering
University of Alabama in Huntsville
Huntsville, Alabama 35899, U.S.A.

Abstract

One of the most fundamental properties of a neural network is its (storage) capacity. It determines the practical usefulness as well as the (storage) limitations of the neural network. So far, the capacity of neural networks has mainly been studied for specific learning rules only.

This paper presents some theoretical upperbounds on the (storage) capacity of neural networks. These generally applicable upperbounds are topology independent and learning rule independent. The problem of capacity is approached from different points of view. An overall upperbound *based on combinatorics*, and a tighter upperbound from information theoretical point of view, are given. Also included is an upperbound for linear neural networks (or *discriminants*).

For general reference an extensive bibliography on the subject of neural network capacity is appended.

Keywords

(artificial) neural networks, connectionism, neural network capacity, neural network statics, neural network connectivity

INTRODUCTION

Much discussion is taking place about the usefulness of (artificial) neural networks. The viability of their use depends to some extent on their limitations. One of the most fundamental limitations is the (storage) capacity of neural networks. One wants to be able to store and process as much information in a neural network as possible. The capacity issue has many impacts on fundamental and applied research on neural networks; cf. (DARPA-88). It is essential for the work on connectivity and optimal topologies of neural networks (Fiesler-90).

Since the information presented to neural networks can be represented as patterns, it is useful to examine the pattern capacity of a neural network. The ***pattern-capacity*** (C) of a neural network is the number of different patterns that can be stored in that neural network, where a ***pattern*** consists of a set of values called ***pixels***, and a pixel can assume any value from a (finite or infinite) set, called ***pixel value set***. A pattern is said to be ***stored*** in a neural network if it can be retrieved (within certain exactness limits) as an output from the network by feeding the corresponding input into the network. In this paper, exact recall of the patterns is assumed, this means that no errors are allowed in the (possible) recall of a pattern.

Neural networks are characterized by their architecture (including their topology) and their learning rules. Up till now, neural network capacity research has been oriented towards neural netwcrks with a specific learning rule (cf. the appended biography on capacity). In order to get an approximation of the capacity of an arbitrary neural network, the concepts which are general to neural networks have to be explored. To remain independent of topology and learning rule, one has to restrict considerations to static entities like the number of neurons in the network (N), the number of weights (W), and in case of discrete neural networks: the number of ***discretization levels*** for the weights (D), and the ***pixel value set cardinality*** (d), which is the number of different (input) values possible for a pattern pixel. A ***discrete neural network*** is defined as a neural network with discrete interconnection strengths (weights). This work is based on discrete neural networks, since all (computer) implementations of theoretically continuous neural networks have a finite precision they can be seen as discrete ones. For example in a computer simulation of a continuous neural network with b bit number representation, the number of discretization levels, and the pixel value set cardinality, can maximally be 2^b. An introduction to discrete neural networks and related definitions are described in (Fiesler-88).

AN UPPER UPPERBOUND

Every weight in a neural network can assume D different values, and there are W weights in the network. Therefore, using plain combinatorics, the number of different patterns that can be represented (this is the number of different distinguishable states) in a neural network is

$$D^W.$$

An input pattern is copied into the activation values of the input neurons. There are N_1 input neurons and each can assume one of the d different possible values. The number of different input patterns that can be presented to a layered neural network is

$$d^{N_1}.$$

In order to store information in a neural network, it has to flow through the input neurons. Hence, the smallest value of the two equations given above, will be the information bottleneck for the network. The upperbound for the pattern-capacity is therefore

$$\text{Minimum}(d^{N_1},\, D^W).$$

Although, since for multi-layered neural networks $N_1 \leq W$, for most networks: $d^{N_1} < D^W$ and in this case the upperbound will be d^{N_1}.

A TIGHTER UPPERBOUND

This number, which is a theoretical capacity upperbound, can be lowered if information-capacities are considered. Information theory defines the total amount of information, or entropy, of a system to be $\sum_{i=1}^{n} p_i \log p_i^{-1}$, where p_i is the probability of occurrence of state i of the network, $1 \leq i \leq n$. (The notation "log" stands for any base logarithm; a convenient choice is base D.) The maximum amount of information is obtained when all p_i's are equal (Hamming-80); i.e. $p_i = n^{-1}$, and the total information is $\log n$. An upperbound for the information-capacity of a neural network is therefore

$$\log D^W = W \log D.$$

Analogous, the information-capacity of an input pattern is

$$\log d^{N_1} = N_1 \log d.$$

If the number of patterns to be stored is C, the total input to the network is then

$$CN_1 \log d.$$

If we let C be the pattern-capacity of the neural network, this quantity has to be equal to the total information in the neural network. Therefore, if we combine both formulas, we get an upperbound for the pattern-capacity of a neural network of

$$\frac{W \log D}{N_1 \log d}.$$

This results, applied to fully ***interlayer connected*** neural networks gives, for a two layer network ($L=2$ and $W=N_1 N_2$):

$$C \leq \frac{N_2 \log D}{\log d},$$

and for an auto-associative neural network ($N_1 = N_L$) which has three layers ($L=3$ and $W=N_2(N_1+N_3)$):

$$C \leq \frac{2N_2 \log D}{\log d},$$

where the pattern-capacity is directly proportional to the number of neurons in the hidden layer, since D and d are constants.

UPPERBOUNDS FOR LINEAR NEURAL NETWORKS

Linear problems are well understood mathematically (Pao-89). The non-linearity of neural networks is what makes them hard. So in order to get a grip on the upperbound of the capacity of neural networks, the non-linearity is stripped for a moment and linear neural networks (with interlayer connections only) are observed. A two layer linear neural network with only interlayer connections is known as a (linear) *discriminant*. Observe a simple linear neural network, where

$$a_{l,j} = \sum_{i=1}^{N_{l-1}} W_{l,ij} a_{l-1,i} \qquad \text{for } 1 \leq j \leq N_l,$$

in which $a_{l,i}$ represents the activation value of neuron i in layer l. For the input layer ($l = 1$), this value is equal to the pixel value for the corresponding input neuron of the network. For a two layer system the system consists of N_2 equations. In these equations the interconnection weights are the variables. In a fully interlayer connected neural network there are $W = N_1 N_2$ weights, and therefore $N_1 N_2$ *degrees of freedom* (independent variables), which means that the system of linear equations is completely determined by giving $N_1 N_2$ variables a value.

When a pattern is presented to the two layer network, the activation values are known and this gives N_2 equations *in* $N_1 N_2$ *unknowns*. The number of degrees of freedom of a system of E linear independent equations in U unknowns is

$$\text{Maximum}(U - E, 0)$$

So in the previous case, the number of degrees of freedom is $N_1 N_2 - N_2 = N_2(N_1 - 1)$. Each new pattern, which has at least some component orthogonal to the other patterns, gives a new set of N_2 equations in the same variables. Thus after P patterns, $N_1 N_2 - P N_2 = N_2(N_1 - P)$ degrees of freedom are left over. A system of linear independent equations is solved when there are no degrees of freedom left. This happens when $P = N_1$. So an upperbound for the pattern-capacity of a two layer neural network when considering linearly independent patterns to be stored completely (error-free) is: N_1.

If we extend this to more layers and assume the activation values to be known, we have to incorporate the other layers as well and get as an upperbound for the pattern-capacity for a multi layer linear neural network:

$$C \leq \sum_{l=1}^{L-1} N_l = N - N_L.$$

SUMMARY

In this paper a number of analytically derived upperbounds on the (storage) capacity of neural networks are presented. They are independent of the network topology and the learning rule used. It is shown that the maximum amount of information that can be stored in an arbitrary neural network is normally limited by the number of input states, which is exponential in the number of input neurons.

For an exact recall, the capacity upperbound can be 'compressed' to an amount which is proportional to $\frac{W}{N_1}$; the total number of weights divided by the number of input neurons of the network. For layered neural networks, with up to three layers, the upperbound becomes linear in N_2, the size of the second layer.

An upperbound for the number of partially orthogonal patterns that can be stored in a linear neural network is proportional to N, the total number of neurons in the network.

ACKNOWLEDGEMENTS

The authors would like to express their thanks to Dr. J. Patrick Ryan for his continuing support, and Drs. Vincent J. Harrand and Ms Indu Saxena for their suggestions to improve this paper.

† This work is partially supported by ONR (contract # N0014-86-K-0591).

REFERENCES

Defense Advanced Research Projects Agency. 1988. *DARPA neural network study*. AFCEA International Press, Fairfax, Virginia, U.S.A., ISBN: 0-916159-17-5.

Fiesler, E.; A. Choudry; and H.J. Caulfield. 1988. "Weight discretization in backward error propagation neural networks". accepted for publication in: *IEEE transactions on systems, man, and cybernetics.*

Fiesler, E.; H.J. Caulfield; A. Choudry; and J. Patrick Ryan. 1990. "Maximum capacity topologies for fully connected layered neural networks with bidirectional connections". submitted to: the IEEE/INNS International Joint Conference on Neural Networks (IJCNN), San Diego, (June 17-21), 1990.

Hamming, Richard W. 1980. *Coding and Information Theory*. Prentice-Hall, Englewood Cliffs, New Jersey, ISBN: 0-13-139139-9.

Pao, Yoh-Han. 1989. *Adaptive Pattern Recognition and Neural Networks*. Addison-Wesley, Reading, Massachusetts, ISBN: 0-201-12584-6.

A BIBLIOGRAPHY ON THE CAPACITY OF NEURAL NETWORKS

Abu-Mostafa, Yaser S. and Jeannine-marie St. Jacques. 1985. "Information capacity of the Hopfield model". *IEEE transactions on information theory*, volume IT-31, number 4 (July) pages 461-464.

Amit, Daniel J.; Hanoch Gutfreund; and H. Sompolinsky. 1985. "Storing infinite Number of Patterns in a Spin-Glass Model of Neural Networks". *Physical review letters*, volume 55, number 14, (30 September), pages 1530-1533.

Bak, Chan S. and Michael J. Little. 1988. "Memory Capacity of Artificial Neural Networks with High Order Node Connections". *Proceedings of the IEEE international conference on neural networks 1988*, volume I, pages 207-216, (San Diego, California, June 23-27, 1988). SOS Printing, San Diego, California, IEEE Catalog Number 88CH2632-8.

Baldi, Pierre and Santosh S. Venkatesh. 1987. "Number of Stable Points for Spin-Glasses and Neural Networks of Higher Orders". *Physical review letters*, volume 58, number 9, (2 March), pages 913-916.

Baldi, Pierre and Santosh S. Venkatesh. 1986. "On properties of networks of neuron-like elements". *Neural Information Processing Systems, Proceedings of the first IEEE Conference on Neural Information Processing Systems*, pages 41-51, editor: Dana Z. Anderson, (Denver, Colorado, November 8-12, 1985). American Institute of Physics, New York, New York, ISBN: 0-88318-569-5.

Baldi, Pierre. 1988. "Neural networks, Orientations of the Hypercube, and Algebraic Threshold Functions". *IEEE transactions on information theory*, volume IT-34, number 3, (May), pages 523-530.

Chiueh, Tzi-Dar and Rodney M. Goodman. 1988. "High-Capacity Exponential Associative Memories". *Proceedings of the IEEE international conference on neural networks 1988*, volume I, pages 153-160, (San Diego, California, June 23-27, 1988). SOS Printing, San Diego, California, IEEE Catalog Number 88CH2632-8.

Griffith, J.S. 1965. "Information Theory and Memory". *Molecular biophysics, Proceedings of an International Summer School*, pages 411-435, editors: Bernard Pullman and Mitchel Weissbluth, (Squaw Valley, California, August 17-28, 1964). Academic press, New York, New York, LoCCCN 65-23843.

Hopfield, John J. 1982. "Neural networks and physical systems with emergent collective computational abilities". *Proceedings of the National Academy of Sciences of the U.S.A.*, volume 79, (April), pages 2554-2558.

Keeler, James D. 1987. "Comparison of Information Capacity of Hebbian Neural Networks". *Proceedings of the IEEE first international conference on neural networks*, volume III, pages 253-260, editors: Maureen Caudill and Charles Butler, (San Diego, California, June 20-24, 1987). SOS Printing, San Diego, California, IEEE Catalog Number 87TH0191-7.

Kinser, Jason M. and H. John Caulfield. "Error Correcting Network". 1989. presented at: the IEEE/INNS International Joint Conference on Neural Networks (IJCNN 89), (Washington D.C., June 18-22, 1989). abstract in: *Proceedings of the IJCNN 89*, volume II, page 570, IEEE TAB neural network committee / SOS Printing, San Diego, California, IEEE Catalog Number 89CH2765-6.

Kleinfeld, D. and D.B. Pendergraft. 1987. ""Unlearning" increases the storage capacity of content addressable memories". *Biophysics journal*, volume 51, (January), pages 47-53.

Kuh, Anthony and Bradley W. Dickinson. 1989. "Information Capacity of Associative Memories". *IEEE transactions on information theory*, volume IT-35, number 1, (January), pages 59-68.

Little, W.A. and Gordon L. Shaw. 1978. "Analytic Study of the Memory Storage Capacity of a Neural Network". *Mathematical Biosciences*, volume 39, pages 281-290.

Little, Michael J. and C.S. Bak. 1986. "Enhanced Memory Capacity of a Hopfield Neural Network". *Real Time Signal Processing IX*, SPIE, volume 698, pages 150-156.

Marom, E.; B.H. Soffer; and U. Efron. 1987. "On the Capacity of Outer Product Associative Memories". abstract: *Journal of the Optical Society of America A*, (special edition 1987 annual meeting), Optics and Image Science, volume 4, number 13, page 131.

McEliece, Robert J. and Edward C. Posner. 1985. "The number of stable points of an infinite-range spin glass memory". *JPL Telecomm. and Data Acquisition Progress Report*, volume 42, number 83, pages 209-215.

McEliece, Robert J.; Edward C. Posner; Eugene R. Rodemich; and Santosh S. Venkatesh. 1987. "The Capacity of the Hopfield Associative Memory". *IEEE transactions on information theory*, volume IT-33, number 4, (July), pages 461-482.

Neumann, John von. 1958. *The Computer and the Brain*. Yale University Press, New Haven, Connecticut.

Orponen, Pekka. 1989. "An Experimental Evaluation of the Optimal Capacity of the Hopfield Associative Memory". presented at: the IEEE/INNS International Joint Conference on Neural Networks (IJCNN 89), (Washington D.C., June 18-22, 1989). abstract in: *Proceedings of the IJCNN 89*, volume II, page 612, IEEE TAB neural network committee / SOS Printing, San Diego, California, IEEE Catalog Number 89CH2765-6.

Peretto, P. and J.J. Niez. 1986. "Long-term memory storage capacity of multiconnected neural networks". *Biological Cybernetics*, volume 54, pages 53-63.

Saad, D. and E. Marom. 1989. "Capacity expansion of (the Hopfield) neural networks model using external coding". presented at: the IEEE/INNS International Joint Conference on Neural Networks (IJCNN 89), (Washington D.C., June 18-22, 1989). abstract in: *Proceedings of the IJCNN 89*, volume II, page 571, IEEE TAB neural network committee / SOS Printing, San Diego, California, IEEE Catalog Number 89CH2765-6.

Samad, Tariq and Paul Harper. 1987. "Associative Memory Storage Using a Variant of the Generalized Delta Rule". *Proceedings of the IEEE first international conference on neural networks*, volume III, pages 173-183, editors: Maureen Caudill and Charles Butler, (San Diego, California, June 20-24, 1987). SOS Printing, San Diego, California, IEEE Catalog Number 87TH0191-7.

Venkatesh, Santosh S. 1986. "Epsilon capacity of neural networks". *Neural networks for computing*, pages 440-445, American Institute of Physics (A.I.P.) conference proceedings 151, editor: John S. Denker, (Snowbird, Utah, April 13-16, 1986). American Institute of Physics, New York, New York, ISBN: 0-88318-351-X.

Venkatesh, Santosh S. and Demitri Psaltis. 1989. "Linear and Logarithmic Capacities in Associative Neural Networks". *IEEE transactions on information theory*, volume IT-35, number 3, (May), pages 558-568.

Willcox, Charles R. 1988. "Exponential Storage and Retrieval in Hierarchical Neural Networks". presented at: the first annual International Neural Network Society (INNS), (Boston, Massachusetts, September 6-10, 1988). abstract in: *Neural Networks*, volume 1, Supplement 1, page 232.

Short biography of the presenting author

Emile Fiesler was born in Amsterdam, The Netherlands, in 1961. He received the B.Sc. and M.Sc. degrees in computer science in 1984 and 1986, respectively, from the University of Amsterdam, Amsterdam, The Netherlands. He is currently working towards his Ph.D. degree in computer science at the University of Alabama in Huntsville, Huntsville, Alabama, U.S.A.

Since 1986 he has been a research fellow at the Department of Computer Science of the University of Alabama in Huntsville in collaboration with the Center for Applied Optics.

He is a member if the IEEE. His main interests is in fundamental aspects of artificial neural networks.

Maximum capacity topologies for fully connected layered neural networks with bidirectional connections[ʳ]

Emile Fiesler[1,2], H. John Caulfield[2], Amar Choudry[3],
and J. Patrick Ryan[1]

[1]Department of Computer Science
[2]Center for Applied Optics
[3]Department of Electrical and Computer Engineering
University of Alabama in Huntsville
Huntsville, Alabama 35899, U.S.A.

Abstract

One of the main problems in current (artificial) neural network engineering is the lack of design rules for neural networks, i.e. how many layers and how many neurons per layer to choose for a fully connected layered neural network with bidirectional weights. A theory is developed which optimizes the topology of the neural network to allow a maximum potential storage capacity with a minimum amount of neurons.

Keywords: (artificial) neural networks, connectionism, neural network topology, neural network statics, neural network connectivity, neural network capacity

Introduction

Although the field of artificial neural networks, hereafter called neural networks, is a rapidly growing one, some basic questions remain unanswered. One of the most important problems is how to configure a neural network. Many neural network learning rules apply to (fully connected) layered (first order) neural networks with bidirectional weights (or interconnection strengths). A *bidirectional connection* is a connection that has the same connection strength when used for either forward or backward propagation. (If a neural network uses only unidirectional propagation, the interconnection topology of the neural network is identical to one with unidirectional connections.)

For layered neural networks in general, one needs to determine the number of layers and the number of neurons per layer. Since neural networks are used for processing and storage of information, the 'optimal' topology for a neural network is usually one which allows an optimal (information) storage capacity. Since the interconnection strengths (weights) contain the information of the neural network, the information capacity is proportional to the total number of weights in the network [1]. A fully connected neural network will therefore have a higher information capacity than any other interconnection scheme. However, in layered neural networks there are several types of connections.

Counting Weights

In layered neural networks one can discriminate three classes of connections:

Definition : An *interlayer connection* is a connection between neurons in adjacent layers of the neural network.

Definition : An *intralayer connection* is a connection between neurons of the same layer of the neural network.

Definition : A *supralayer connection* is a connection between neurons that are neither in adjacent layers, nor in the same layer of the neural network.

A sub-class of intralayer connections are self-connections:

Definition : A *self-connection* is a connection which originates and terminates at the same neuron.

A neural network can have all possible connections:

Definition : A *plenary neural network* is a neural network which has all possible interlayer, intralayer, and supralayer connections; in other words it is a 'truly' fully connected neural network.

The total number of weights (W) for a neural network with L layers, which has only interlayer connections (i.e. they have neither intralayer nor supralayer connections) is the sum of all possible connections (between each pair of adjacent layers) in the network. The number of connections between two adjacent layers in a fully *inter*connected network is equal to the product of the number of neurons in each of the layers. In order to get the total number of weights for the complete network, a summation is needed over the layers:

$$W = \sum_{l=2}^{L} W_l = \sum_{l=2}^{L} N_{l-1}\, N_l,$$

where W_l stands for the number of weights between layer $l-1$ and l, and N_l is the number of neurons in layer l. Layer 1 is the input layer and N_1 the number of input neurons.

In the case that the neural network has both interlayer and intralayer connections, a number equal to the number of possible connections within a layer has to be added for each layer. The total number of connections becomes thus:

$$\sum_{l=2}^{L} N_{l-1}N_l + \left[\frac{N_1}{2}(N_1 \pm 1) + \right] \sum_{l=2}^{L} \frac{N_l}{2}(N_l \pm 1) =$$

$$\left[\frac{N_1}{2}(N_1 \pm 1) + \right] \sum_{l=2}^{L} N_l \left(N_{l-1} + \frac{N_l \pm 1}{2}\right),$$

where the part between square brackets is optional; it is deleted when no intralayer connections are present in the input layer. The ±-symbol denotes the option for having self-connections. If self-connections are present, addition has to be used, and substraction otherwise.

The number of weights in a network which, besides interlayer connections also has supralayer connections can be calculated by summing over all the neurons in all the layers, multiplied by the number of neurons in all the layers of a higher index:

$$\sum_{l=2}^{L} N_{l-1} \sum_{m=l}^{L} N_m = \sum_{m=1}^{L-1} \sum_{l=1}^{m} N_l N_{m+1}.$$

When combining the previous two formulas, the total number of neurons is obtained for a plenary neural network, which has all three types of connections:

$$W = \left[\frac{N_1}{2} (N_1 \pm 1) + \right] \sum_{l=2}^{L} \left(\frac{N_l}{2} (N_l \pm 1) + N_{l-1} \sum_{m=l}^{L} N_m \right)$$

where the part in square brackets is again optional and used when intralayer connections are desired in the input layer. In the case that the intralayer connections in the input layer are also present, the formula becomes equal to

$$\sum_{l=1}^{L} \frac{N_l}{2} (N_l \pm 1) + \sum_{l=2}^{L} N_{l-1} \sum_{m=l}^{L} N_m.$$

Since a plenary network can be represented as a fully connected graph, the previous equation is equal to:

$$W = \frac{N}{2} (N \pm 1)\,,$$

(this is the number of edges in a fully connected undirected graph with N vertices); where N is the total number of neurons in the network: $N = \sum_{l=1}^{L} N_l$.

Optimal topologies

Depending on the type(s) of interconnections present, the capacity of a neural network can be optimized by varying the topology of the network. Plenary neural networks are a trivial case; there topology is always optimal, since they can be seen as a fully connected (undirected) graph, whose number of edges only depends on the number of vertices.

For layered neural networks with only interlayer connections (the most used topologies), the configuration topology does make a difference. Let the total number of weights $W = \sum_{l=2}^{L} N_{l-1} N_l$, as defined before. For a two layer neural network $L = 2$, $W = N_1 N_2$, and $N = N_1 + N_2$. The total number of weights, W can be represented as a function of N: $W(N) = N_1 N_2$. Using $N = N_1 + N_2$, W can be transformed into a function of N_1: $W(N_1) = N_1(N - N_1)$. To find the optimal topology, the derivative of $W(N_1)$ with respect to N_1 has to be determined: $\frac{dW(N_1)}{dN_1} = N - 2N_1$. A maximum is found and this gives the optimum:

$$W = \frac{N^2}{4} \text{ at } N_1 = N_2 = \frac{N}{2}.$$

Since the number of neurons and the number of weights are integral numbers, $W = \left\lfloor \frac{N^2}{4} \right\rfloor$ and N_2 can be choosen $\lfloor \frac{N}{2} \rfloor$ or $\lceil \frac{N}{2} \rceil$.

The three layer system $L=3$ and $W=N_2(N_1+N_3)$ gives analogously: $W(N_1, N_2) = W(N_2, N_3) = N_2(N - N_2)$ or $W(N_1, N_3) = N_1(N - N_1 - 2N_3) + N_3(N - N_3)$. Maximization gives a maximum at $N_2 = \frac{N}{2}$ and $N_3 = \frac{N}{2} - N_1$. The maximum for W is again $\left\lfloor \frac{N^2}{4} \right\rfloor$.

For more than three layers, the outcome of the maximization procedure is: drop all but two or three layers, and the same maximum holds; in other words multi-layer systems (>3) layers are not optimal. This outcome coincides with the neural network interpretation of Kolmogorov's theorem, which states that the capabilities of a neural network with more than three layers does not exceed the capabilities of a three layer neural network with $2N_1+1$ neurons in the hidden layer and only interlayer connections [2].

For neural networks with interlayer plus intralayer connections, a fully connected two layer network is equal to a two layer plenary neural network. It has $W = \frac{(N_1+N_2)(N_1+N_2+1)}{2} = \frac{N}{2}(N+1)$ weights. So there is no absolute maximum; any distribution of the neurons over the two layers gives this "maximum". For more than two layers the outcome of the optimization is: drop all but one or two layers and distribute the neurons over these layers. The maximum W is therefore the same as for the plenary neural network.

Layered neural networks with interlayer and supralayer connections have a different optimum: Since two layer neural networks do not have supralayer connections, the smallest networks to study here are three layer networks: $L = 3$ and $W = N_1N_2 + N_1N_3 + N_2N_3$. W can be written as a function of two variables again: $W(N_1, N_2) = N_1(N - N_1 - N_2) + N_2(N - N_2)$, $W(N_1, N_3) = N_3(N - N_1 - N_3) + N_1(N - N_1)$, or $W(N_2, N_3) = N_2(N - N_2 - N_3) + N_3(N - N_3)$. Maximization gives a maximum at $N_1 = N_2 = N_3 = \frac{N}{3}$. The maximum for W is $\frac{N^2}{3}$. This can be generalized and proven for any number of layers. The maximum is found at $N_l = \frac{N}{L}$, for $L > 2$ and $1 \leq l \leq L$, and the maximum is

$$W = \frac{N^2(L-1)}{2L}.$$

Thus in the case of both interlayer and superlayer connections: Since the number of neurons is a positive integer, each layer gets at least $\lfloor \frac{N}{L} \rfloor$ neurons, and the rest of the neurons $(N - L \lfloor \frac{N}{L} \rfloor)$ can be distributed over the layers. The number of weights is also a positive integer. The floor function can only be applied for neural networks with less than eight layers, since the maximum deviation between the optimal and the actual number of weights can be as large as $\frac{L}{8}$ 'weights'.

Conclusions

The optimal topology and maximum number of connections for all the interconnection schemes are given in this table:

interconnection	*optimal topology*			
structure	L_{min}	L_{max}	N_l's	W_{max}
inter	2	3	$N_2 = \frac{N}{2}$	$\frac{N^2}{4}$
inter & intra	1	2	any distribution	$\frac{N(N+1)}{2}$
inter & supra	3	no max.	$\forall l : N_l = \frac{N}{L}$	$\frac{N^2(L-1)}{2L}$
plenary	1	no max.	any distribution	$\frac{N(N+1)}{2}$

Acknowledgements

The authors like to thank Dr. Mou-Hsiung Chang, and Indu Saxena for their support, useful remarks, and proofreading of this paper.

† This work is partially supported by ONR (contract # N0014-86-K-0591).

References

[1] Emile Fiesler,
Formal aspects of neural networks,
To be published, 1990.

[2] Robert Hecht-Nielsen,
Kolmogorov's Mapping Neural Network Existence theorem,
from: Proceedings of the IEEE first international conference on neural networks,
volume III, pages 11-14,
editors: Maureen Caudill and Charles Butler,
at: San Diego, California, June 20-24, 1987,
SOS Printing, San Diego, California, 1987,
IEEE Catalog Number 87TH0191-7.

RAPID COMMUNICATIONS

This section was established to reduce the lead time for the publication of Letters containing new, significant material in rapidly advancing areas of optics judged compelling in their timeliness. The author of such a Letter should have his manuscript reviewed by an OSA Fellow who has similar technical interests and is not a member of the author's institution. The Letter should then be submitted to the Editor, accompanied by a LETTER OF ENDORSEMENT FROM THE OSA FELLOW (who in effect has served as the referee and whose sponsorship will be indicated in the published Letter), A COMMITMENT FROM THE AUTHOR'S INSTITUTION TO PAY THE PUBLICATION CHARGES, and the signed COPYRIGHT TRANSFER AGREEMENT. The Letter will be published without further refereeing. The latest Directory of OSA Members, including Fellows, is published in the July 1986 issue of Optics News.

Fundamental speed limitations on parallel processing

Joseph Shamir

When this work was done the author was with the Center for Applied Optics of the University of Alabama in Huntsville; he has now returned to Technion—Israel Institute of Technology, Department of Electrical Engineering, Haifa 32000, Israel.

Received 2 February 1987.
Sponsored by John Walkup, Texas Tech University.
0003-6935/87/091567-01$02.00/0.

The unlimited operating rate of parallel processing systems as suggested by various proposed architectures is questionable. Like the limitations imposed by the Von Neuman bottleneck on serial processing it appears that we also have a fundamental limitation on the possible ultimate speed of parallel processors.

Accepting the fact that the universal speed limit is the velocity of light, we may estimate the time for performing a single processing step on a planar array of $n \times n$ signal elements. The two planes P_1 and P_2 in Fig. 1 represent optical elements (transparencies, lenses, holograms, spatial light modulators, acoustooptic cells, etc.). To perform any processing operation light must be propagated between these two planes to interconnect all the signal elements of one plane with those of the second plane. These interconnections may be achieved by waveguides or, globally, by diffraction. In any case the processing time will be limited by the transit time of light between these two planes. Furthermore, there will also be a skew—a differential time delay between the various interconnection paths (for example, paths 1 and 2 in the figure).

To estimate the time delays involved in the processing we consider free space propagation, denote the distance between the two planes by R, and denote the operating aperture diameter by D. Assuming diffraction-limited resolution, the diameter of each signal element (pixel) may be given by the diffraction-limited spot size,

$$a = 2.44\lambda R/D = 2.44\lambda(f/\text{No.}), \tag{1}$$

where (f/No.) is the f/number of the optical system and λ is the illuminating wavelength. Thus for our $n \times n$ array we shall need an aperture size,

$$D = na = 2.44n\lambda(f/\text{No.}). \tag{2}$$

The maximum delay time is induced on a diagonal trajectory,

$$T_{\max} = \frac{1}{c}(R^2 + D^2)^{1/2}, \tag{3}$$

where c is the velocity of light, while the minimal time is

$$T_{\min} = R/c. \tag{4}$$

These two time delays may be expressed by the f/No. using Eqs. (1) and (2):

$$T_{\max} = \frac{2.44}{\nu} n(f/\text{No.})[1 + (f/\text{No.})^2]^{1/2}, \tag{5}$$

$$T_{\min} = \frac{2.44}{\nu} n(f/\text{No.})^2, \tag{6}$$

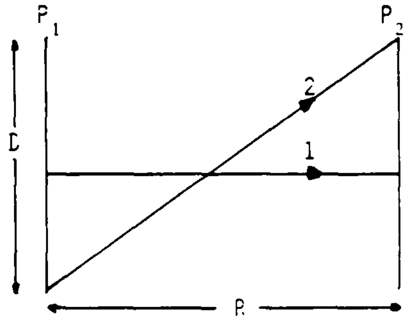


Fig. 1. General building block of an optical system: two planes with separation R and effective aperture D interconnected by propagating light. The two rays represent maximal and minimal path length in a diffractive interconnection.

where ν is the frequency of the illuminating light. From the last two equations one may also derive the skew:

$$t = T_{\max} - T_{\min} = \frac{2.44}{\nu} n(f/\text{No.})^2\{[1 + (f/\text{No.})^{-2}]^{1/2} - 1\}. \tag{7}$$

To get an idea about the magnitude of these time delays let us assume visible illumination with a frequency of 5×10^{14} Hz, an f/No. of 2, and an array of the order of a TV frame with $n = 500$. Substitution of these numbers results in $T_{\min} = 10$ ps and $t = 1.2$ ps. In an actual system involving a number of processing planes and possibly waveguides, these numbers may have to be multiplied by an appreciable factor. For example, a simple optical correlator ($4f$ system) has a factor of 4 leading to a differential delay of 4.8 ps with a total delay of 40 ps.

The above time delays are quite small compared to current processing facilities and presently available device responses. However, considering proposed operation with femtosecond pulses these delays may become the ultimate limiting factors. The overall time delay must not concern pipelined systems but it may become quite important in complex architectures such as those involving feedback loop operations. Equation (7) indicates that the overall time delay increases with increasing f/No. while the skew approaches the limit $1.22n/\nu$. Thus these effects should be taken into consideration for very high speed architectures. For example, by using optical fibers or other guiding elements one may solve the skew problem but by doing this the pipelining delays will be increased.

In conclusion, we note that the limiting time factors were estimated for thin optical elements in free space. The fact that the vacuum velocity of light is a universal speed limit may indicate that we are dealing here with a universal bottleneck influencing all possible approaches to parallel signal processing. This bottleneck is proportional to the operating wavelength suggesting that computing with visible or IR light may be just an intermediary step toward an even more advanced approach.

The author is pleased to thank J. F. Walkup for reading the manuscript and making some useful comments.

This work was partially supported by the Office of Naval Research under contract N00014-86-K-0591.

APPENDIX E

OPTICAL SYSTOLIC ARRAY PROCESSING USING A NOVEL INTEGRATED ACOUSTO-OPTIC MODULATOR MODULE

Technical Report
for
Office of Naval Research
N00014-86-K-0591

by
Dr. Chen S. Tsai
Tsai Associates, Inc.
5702 Highgate Terrace
Irvine, CA 92715

February 5, 1990

I. OBJECTIVE

The objective of this research is to advance the performance characteristics and applications of compact integrated acousto-optic and acousto-electro-optic Bragg modulator modules. The following specific research task were proposed and pursued:

1. Analysis on Acousto-optic Bragg Diffraction in Channel-Planar Waveguide
2. Identification of Existing and New Architectures and Algorithms
3. Comparison between Acousto-optic and Electro-optic Modulation/ Multiplication Schemes

During the course of this research significant progress was made in each task.

II. ACCOMPLISHMENTS

A summary of accomplishments on each task now follows:

1. Analysis On Acousto-optic Bragg Diffraction In Channel-Planar Waveguide

Fig. 1 shows the configuration of the integrated acousto-optic (AO) Bragg modulator module[1] that has been analyzed. An array of light beams coupled into the channel-waveguide array at the input endface of the $LiNbO_3$ channel-planar composite waveguide are expanded and collimated by the titanium-indiffused proton-exchanged (TIPE) waveguide lens array[2] before incidence upon the surface acoustic waves (SAW) generated by the interdigital SAW transducer. The array of Bragg-diffracted light beams are then collected and focused upon the output endface of the composite waveguide by the large-aperture TIPE lens. By varying the carrier frequency of the rf driving signal the Bragg-diffracted light beams are scanned along the output endface.

At the outset a potential distinction between the AO interaction geometry under consideration and the conventional one that involves a single SAW and a single light beam in a purely planar waveguide substrate[3] was identified. This potential distinction was based on the fact that optical anisotropy and the very small aperture of the multiple incident light beams (for example, 5 to 10μm) and thus the resulting spreading of the light beams (by diffraction) will significantly affect the performance characteristics of the device module. However, a subsequent numerical calculation shows that to a good approximation the spreading angle can be determined using a conventional formula and since the microlens array is placed at a short distance from the output edges of the channel-waveguide array, no significant effect through optical anisotropy has been concluded. Accordingly, it has been concluded that the ultimate performance characteristics of the integrated AO Bragg modulator module such as diffraction efficiency, rf bandwidth, rf drive power, nonlinearity, and dynamic range are practically identical to that of a conventional AO Bragg modulator in a planar waveguide[3].

2. Identification Of Existing And New Architectures And Algorithms

The integrated AO Bragg modulator module of Fig. 1 was found to be rather inconvenient and limited in applications such as matrix-vector and matrix-matrix multiplications as one set of input data must be used to modulate the input light beams. This is so because laser arrays (such as diode laser arrays) with capability for independent modulation of each individual laser are not commercially available. Consequently, much efforts were made to identify and explore other new architectures.

The two device architectures that have been identified and explored are shown in Figs. 2 and 3. The basic architecture common to both modules is a composite waveguide in which a channel-waveguide array, a planar waveguide, a linear TIPE microlens array, Bragg modulator arrays, and a large-aperture TIPE lens are integrated in a common $LiNbO_3$ substrate. The channel waveguide array (only four elements are shown) is aligned wth the linear microlens array. The two device modules presented in this report utilizes, respectively, a herringbone Bragg electrode array (Fig.2) and a SAW transducer and conventional Bragg electrode array combination (Fig. 3). The microlens array was used to capture, expand, and collimate the multiple light beams from the channel-waveguide array before their incidence upon the resulting electro-optic (EO) and AO-EO Bragg diffraction gratings, while the large-aperture lens collects and focuses the multiple Bragg-diffracted light beams upon a photodetector. In operation, "multiplication" of data is carried out by the Bragg modulators, while "addition" of the resulting products by the large-aperture integrating lens.

Since this particular program had not provided any funds for fabrication and testing of the two device modules, actual design, fabrication and testing were subsequently carried out through other programs. Some of the experimental results have been published[4,5].

In summary, the two device architectures identified and explored have been shown to be capable of realization of high-packing density multichannel integrated optic modules with applications to data processing and computing including programmable correlation of binary sequences[6].

3. Comparison Between Acousto-optic And Electro-optic Modulation/ Multiplication Schemes

A. Acousto-optic Modulation/Multiplication Scheme

Efficient and wideband AO Bragg diffraction by the SAW was achieved in the integrated AO-EO modules. In contrast to their EO counterparts these integrated AO modules have the unique capability to input the data in a pipeline fasion via the SAW. Since the number of operations per second increases with the number of input light beams it is desirable to design and fabricate large arrays of channel waveguides and microlenses with as small an aperture as possible. Using 60 μm as the aperture of the linear microlens array the possibl. number of the light channels will be as large as 333 for a SAW propagating path of 2.0 cm. Since the velocity of a Z-propagating SAW in Y-cut $LiNbO_3$ is $3.5x10^5$ cm/sec the corresponding flow rate for the data is approximately 60 MHz. Naturally, if the aperture of each microlens element is reduced to 30 μm both the number of light channels and the data flow rate will be increased by a factor of two. A specific application of the IO module to optical systolic array processing and computing[7], namely, matrix-vector multiplication was successfully carried out.

B. Electro-optic Modulation/Multiplication Scheme

As shown in Fig. 2, the integrated EO Bragg modulator module results by replacing the SAW-generated AO diffraction grating with an array of EO Bragg diffraction gratings that were created by applying voltages across an array of interdigital finger electrodes. Efficient and wideband Bragg diffraction have been achieved using the electrode arrays with 13 μm periodicity and 2.0 mm aperture. Specifically, 95% diffraction at a drive voltage of 6.0 volt and 870 MHz rf bandwidth were measured[4]. It is important to note that the two separate electrode arrays of the Herringbone type facilitate application and thus multiplication of two independent sets of data. Thus, in contrast to their AO counterparts, these integrated EO modules can accept multiple sets of data as well as at a much higher rate than is possible with the SAW. This capability has been utilized to perform matrix-matrix multiplication[6].

III. RELATED PUBLICATIONS

1. P. Le, D.Y. Zang, C.S. Tsai, "Integrated Electro-optic Bragg Modulator Modules for Matrix-Vector and Matrix-Matrix Multiplications," Appl. Opt., 27, 1780-1785 (May 1988)

2. C.S. Tsai, "Integrated Optical Device Modules in $LiNbO_3$ for Computing and Signal Processing," J. Modern Optics, 35, 965-977 (June 1988)

3. C.S. Tsai, D.Y. Zang, and P. Le, "High-Packing-Density Multichannel Integrated-Optic Modules in $LiNbO_3$ for A Programmable Correlation of Binary Sequences," Opt. Letts., 14, 889-891 (Aug. 1989)

IV. REFERENCES

1. C.S. Tsai, D.Y. Zang, and P. Le, Appl. Phys. Lett., 47, 549 (1985)
2. D.Y. Zang and C.S. Tsai, Appl. Phys. Lett., 46, 703 (1985)
3. C.S. Tsai, IEEE Trans. Circuits Syst. CAS-26, 1072 (1979)
4. P. Le, D.Y. Zang, C.S. Tsai, Appl. Opt., 27, 1780-1785 (1988)
5. C.S. Tsai, J. Modern Optics, 35, 965-977 (1988)
6. C.S. Tsai, D.Y. Zang, and P. Le, Opt. Lett., 14, 889-891 (1989)
7. H.J. Caulfield, W.J. Rhodes, M.J. Foster, and S. Horvitz, Opt. Commun. 40, 86 (1981)

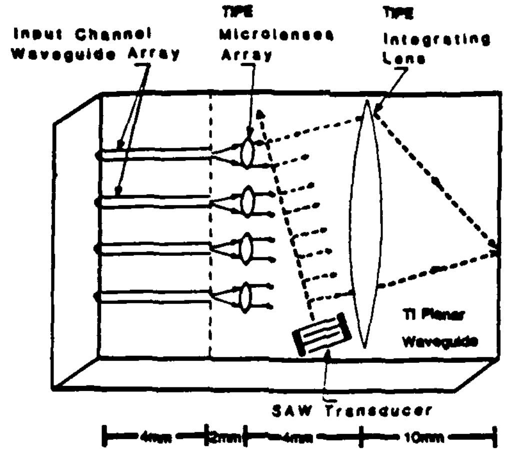


Fig. 1 An Integrated Acoustooptic Bragg Modulator Module Using A Channel-Planer Composite Waveguide in $LiNbO_3$

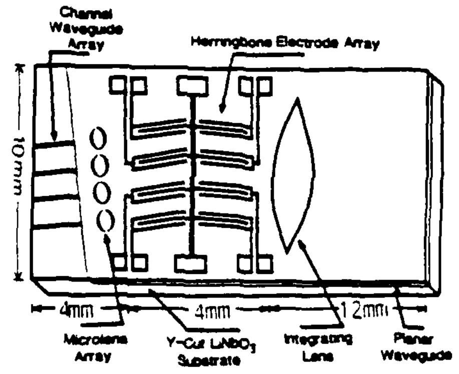


Fig. 2 Multichannel integrated EO module in Y-cut $LiNbO_3$.

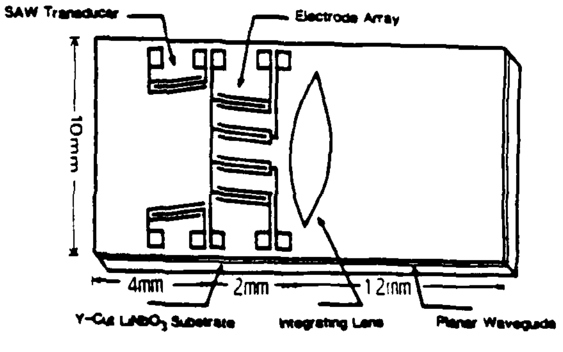


Fig. 3 Multichannel integrated AO-EO module in Y-cut $LiNbO_3$.

APPENDIX F

POLYNOMIAL CONVOLUTION ALGORITHM FOR MATRIX MULTIPLICATION WITH APPLICATION FOR OPTICAL COMPUTING

Richard Barakat
John Reif

Division of Applied Sciences
Harvard University
Cambridge, Massachusetts 02138

1. INTRODUCTION

It is generally agreed that in the realm of computational linear algebra, particularly the multiplication of two matrices, optical computing has an inherent speed of execution advantage over digital electronics (but see Section 4). Investigators in optical computing, have generally taken matrix multiplication algorithms directly from the mathematical literature and modified them for use in optical computing, some representative papers are [1–4]. Alternately optical architectures have been developed to carry out such computations, e.g. [5–11].

One purpose of the present communication is to describe our polynomial convolution algorithm which is an *ab initio* development of matrix multiplication for use in optical computing. A second purpose is to consider the situation where the matrices are$\overset{\text{so}}{\wedge}$large that they cannot be stored *simultaneously* on optical masks (hereafter termed the storage problem). As we will show in Section 4, the speed advantage of the methods advocated in [1–4] are compromised because the matrix elements are not equally accessible. Furthermore, we make plausible that the polynomial convolution algorithm is robust with respect to this debilitating situation in that it is still possible to obtain a reasonable concurrency over the more classical algorithms because of the simplified bookkeeping and modular structure of the convolution algorithm.

2. POLYNOMIAL CONVOLUTION ALGORITHM

In view of the initial complexity of the algorithm we proceed in three stages. In the first stage we give the explicit expressions and verify these formulae in the second stage. Finally, we outline a construction which leads to the various formulas.

We begin by considering the matrix product $\mathbf{C} = \mathbf{AB}$ where $\mathbf{A}$ is of the size $n_1 \times n_2$, $\mathbf{B}$ is of size $n_2 \times n_3$, and $\mathbf{C}$ is of size $n_1 \times n_3$, with corresponding matrix elements: a_{ij}, b_{jk}, and c_{ik}. Let x be an indeterminate, and associate with $\mathbf{A}$ and $\mathbf{B}$ the polynomials $P(x)$ and $Q(x)$

$$P(x) = \sum_{s=0}^{(n_1-1)n_2n_3+n_2-1} p_s x_s \tag{2.1}$$

$$Q(x) = \sum_{t=0}^{n_2n_3-1} q_t x^t \quad . \tag{2.2}$$

Note that the degree of $P(x)$ is $(n_1 - 1)n_2n_3 + n_2 - 1$ which involves not only the size of $\mathbf{A}$ through n_1 and n_2 but also the size of $\mathbf{B}$ through n_3. The degree of $Q(x)$ is $n_2n_3 - 1$ and only involves the size of $\mathbf{B}$, namely n_2 and n_3. The p and q coefficients are related to the matrix elements of $\mathbf{A}$ and $\mathbf{B}$ by

$$\begin{aligned} p_s &= a_{ij}, \qquad \text{if} \quad s = (i-1)n_2n_3 + j - 1 && (2.3a) \\ &= 0, \qquad \text{if} \quad (i-1)n_2n_3 + n_2 \leq s \leq in_2n_3 && (2.3b) \end{aligned}$$

and

$$\begin{aligned} q_t &= b_{jk}, \qquad \text{if} \quad t = kn_2 - j && (2.4a) \\ &= 0, \qquad \text{if} \quad t \geq n_2n_3 && (2.4b) \end{aligned}$$

with: $1 \leq i \leq n_1$, $1 \leq j \leq n_2$ and $1 \leq k \leq n_3$.

We claim that the elements of the matrix product $\mathbf{C}$ are given by selected coefficients of the polynomial

$$\begin{aligned} R(x) &= P(x)Q(x) \\ &= \sum_{m=0}^{n_1 n_2 n_3 - 1} r_m x^m \end{aligned} \tag{2.5}$$

where

$$r_m = \sum_{s=0}^{m} p_s q_{m-s} \tag{2.6}$$

is the discrete convolution of the p and q coefficients. These selected r_m are given by

$$r_m = c_{ik} \;, \qquad \text{if} \quad m = (i-1)n_2 n_3 + kn_2 - 1 \quad . \tag{2.7}$$

A formal proof (which is really a verification of the formulae) is now given. We begin by rewriting. Eq (2.6) in the form

$$r_m = \sum_{s} p_s q_{m-s} = \sum_{\alpha,\beta,\gamma,\delta} a_{ij} b_{jk} \tag{2.8}$$

where the summation in the second series is over:

$$\alpha: \qquad s = (i-1)n_2 n_3 + j - 1 \tag{2.9a}$$

$$\beta: \qquad (i-1)n_2 n_3 < s < (i-1)n_2 n_3 + n_2 \tag{2.9b}$$

$$\gamma: \qquad t = m - s = kn_2 - j \tag{2.9c}$$

$$\delta: \qquad t < n_2 n_3 \quad . \tag{2.9d}$$

The α term is simply Eq. (2.3a), while the β term is the negation of Eq. (2.3b). The γ term follows from Eq. (2.4a), while the δ term is the negation of Eq. (2.4b). Upon substitution of the α term into the β inequality we immediately see that this can only be true

$$1 \leq j \leq n_2 \quad . \tag{2.10}$$

In like fashion, substitution of the γ term into the δ inequality leads to the requirement that

$$m = (i-1)n_2n_3 + kn_2 - 1 \tag{2.11}$$

which is Eq. (2.7). Thus the formulae are verified.

A construction which leads to the various formulae for p_s and q_t in terms of a_{ij} and b_{jk}, respectively uses row vectors. Consider a row vector $\mathbf{p}$ whose elements we denote by p_s (coefficients of the polynomial $P(x)$) composed of the matrix elements a_{ij} of $\mathbf{A}$ and strings of zeros as depicted in Fig. 1A. The range of s is

$$0 \leq s \leq n_1n_2n_3 - n_2n_3 + n_2 - 1 \tag{2.13a}$$

consequently

$$\begin{aligned} p_s &\equiv 0\,, \qquad \text{if} \quad s \geq (n_1-1)n_2n_3 + n_2 && (2.13a)\\ &\equiv 0\,, \qquad \text{if} \quad s \leq in_2n_3 \quad . && (2.13b)\end{aligned}$$

Furthermore the p_s are related to the a_{ij} as given by Eq. (2.3a), as the reader can verify by construction.

In like fashion, we construct another row vector **q** with elements q_t according to Fig. 1B. Unlike **p**, **q** has no strings of zero elements. The range of t is

$$0 \leq t \leq n_2 n_3 - 1 \tag{2.14}$$

so that

$$q_t \equiv 0 \, , \qquad \text{if} \quad t \geq n_2 n_3 \quad . \tag{2.15}$$

Within the range of t, the q_t are related to the b_{jk} by

$$q_t = b_{jk} \, , \qquad \text{if} \quad t = (k-1)n_2 + n_2 - j \tag{2.16}$$

which reduces to Eq. (2.4a).

As an illustrative example of the algorithm, consider the case where **A** is 2 × 2, **B** is 2 × 3 so that **C** is 2 × 3 (*i.e.*, $n_1 = 2$, $n_2 = 2$, $n_3 = 3$). The upper limits on the polynomials P, Q and R are 7, 5, and 11, respectively. The p_s, q_t and r_m coefficients evaluated according to Eqs. (2.3), (2.4) and (2.7) are listed in Table 1. Upon carrying out the convolution operation, Eq. (2.6), in conjurction with this table we have:

$$r_1 = c_{11} = p_0 q_1 + p_1 q_0 = a_{11} b_{11} + a_{12} b_{21} \tag{2.17a}$$

$$r_3 = c_{12} = p_0 q_3 + p_1 q_2 = a_{11} b_{12} + a_{12} b_{22} \tag{2.17b}$$

$$r_5 = c_{13} = p_0 q_5 + p_1 q_4 = a_{11} b_{13} + a_{12} b_{23} \tag{2.17c}$$

$$r_7 = c_{21} = p_6 q_1 + p_7 q_0 = a_{21} b_{11} + a_{22} b_{21} \tag{2.17d}$$

$$r_9 = c_{22} = p_6 q_3 + p_7 q_2 = a_{21} b_{12} + a_{22} b_{22} \tag{2.17e}$$

$$r_{11} = c_{23} = p_6 q_5 + p_7 q_4 = a_{21} b_{13} + a_{22} b_{23} \tag{2.17f}$$

These are, of course, the matrix elements as obtained by more standard procedures.

This completes our description of the algorithm.

3. IMPLEMENTATION AND PARALLELISM OF ALGORITHM

In spite of the complicated looking nature of the algorithm, its *implementation* in optical computing can be carried out in straightforward fashion.

Examination of Fig. 1A shows that the matrix elements a_{ij} of **A** coded into the vector **p** consists of the *rows* of **A** in which strings of zeros are interspaced. Thus all we need to do to handle **A** in this algorithm is to store it on an optical mask according to Fig. 1A. The vector **q** containing the matrix elements b_{jk} is simply the *columns* of **B** in *reverse order*, see Fig. 1B. Obviously we need only code **B** as per Fig. 1B on an optical mask for this aspect of the implementation. Given that both these operations have been carried out we then proceed according to the various formulae quoted in the previous section.

The parallelism of the algorithm (assuming that all the matrix elements of **A** and **B** can be stored in primary storage) manifests itself through the corresponding **p** and **q** vectors. This is best seen by examination of Table 1; the first two components of **p** (*i.e.*, a_{11} and a_{12}) can then be combined simultaneously with (b_{21}, b_{11}), (b_{22}, b_{12}), and (b_{23}, b_{13}) of the vector **q**. While these operations are being carried out, the last two (nonzero) elements of **p** (*i.e.*, a_{21} and a_{22}) are to be combined with (b_{21}, b_{11}), (b_{22}, b_{12}), (b_{23}, b_{13}). Thus we are able to carry out the manipulations leading to the six matrix elements of **C** simultaneously. The general case of two rectangular matrices does not require any detailed comment. Consequently, the polynomial convolution algorithm is at least as fast as the methods advocated in [2, 4] under the *assumed conditions of equally accessible matrix elements.*

4. INFLUENCE OF STORAGE PROBLEM ON ALGORITHM PARALLELISM IN MATRIX MULTIPLICATION

Although the issue of matrix multiplication, in the context of optical computing, has been cast as one of speed of execution of manipulations, this is only one aspect of the problem as we will now see. Realistic signal processing requirements demand very large matrices in order to achieve the resolutions necessary to fulfill the desired goals. Because such large matrices are needed we must study the effect of storage (that is, the extent to which all matrix elements in the two matrices under multiplication are not equally accessible) on the inherent parallelism, and hence speed, of the various algorithms proposed.

When the matrices are small (for convenience we will let them both be square and of size $n \times n$), the entire arrays containing the matrix elements of $\mathbf{A}$ and $\mathbf{B}$ can reside simultaneously in primary storage in the form of matrix masks as described in Goodman [12], then it is possible to carry out all of the manipulations such as described in the algorithms promulgated in [1-4]. Under the small n regime, it is essentially true that all matrix elements are equally accessible. In fact, all the papers that we have succeeded in locating on matrix multiplication (via optical computing) tacitly make the assumption that all matrix elments are equally accessible, *independent of* n.

Let us consider, for example, the inner, intermediate, and outer product methods for the multiplication of matrices. Reference is made to Appendix A for the development of an efficient formalism that yields these representations. Examination of these representations reveals that it is possible to perform the matrix-matrix product at two levels of parallelism. At the first level, the intermediate product methods speed up the execution over the inner product method by a factor of n. At the second level, the outer product method achieves a factor of n^2 over the inner product method. In fact, there are n parallel multiplications and $(n - 1)$ parallel additions to be performed, rather than the n^3 sequential multiplications and $n^3 - n^2)$ sequential additions required at the original element level algorithm. Unfortunately when n is large, the entire arrays cannot reside in primary storage, but only portions thereof. This means that the speed advantage of the outer product method is now lost when computing large matrices, because the matrix elements are not equally accessible! A second tacit assumption is that all arithmetical operations of the same type are equivalent both in cost and in accuracy. This too is violated when n is large.

Thus we cannot simply dismiss the use of the intermediate product representations when n is large. To improve the efficiency of the computation in this situation, it is necessary to maximize the use that is made of the matrix element data on a given matrix mask (containing parts of $\mathbf{A}$ or $\mathbf{B}$) while it is in primary storage. It is probably advantageous to store matrix elements by columns. This is precisely what the column intermediate representation does: $\mathbf{Ce}_j$ is formed as a linear combination of $\mathbf{Ae}_k$ with combination coefficient drawn from $\mathbf{Be}_j$. Obviously one can choose to stow rows so that the row intermediate representations is appropriate. In this scenario, we can only achieve a factor of n in the parallelism in order to accommodate the storage problem. There is also the bookkeeping question as to efficient storage and subsequent manipulation of the matrix elements in accordance with the particular algorithm requirements. Reference is made to Hockney and Jesshope [13] for an overview of such considerations in digital electronic computers.

One possible solution for increasing parallelism when n is large via partitioning. The idea is certainly not new as witness the recent paper by Caulfied *et al.* [3] who choose to use 2×2 matrices for the partitioning. Another viable approach, using the formalism of Appendix A, is the following. Suppose that $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ are partitioned into submatrices. This means that the partitioning of the rows of $\mathbf{A}$ and those of $\mathbf{C}$ is the same, that the partitioning of the columns of $\mathbf{B}$ and those of $\mathbf{C}$ is the same, and that the partitioning of the columns of $\mathbf{A}$ and of the rows of $\mathbf{B}$ is the same. The matrix product can then be formed blockwise. The foregoing remains valid if transcribed by replacing $\mathbf{e}_i$ by $\mathbf{E}_i$, *etc.* $\mathbf{E}_i$ is the i-th block column of the appropriately partitioned identity matrix: the appropriate partitioning being that which is symmetric with respect to rows and columns for the matrix multiplication in question. Consequently, we recognize $\mathbf{AE}_j$

as the j-th block column of $\mathbf{A}$, $\mathbf{E}_i^+\mathbf{A}$ as the i-th block row of $\mathbf{A}$, and $\mathbf{E}_i^+\mathbf{A}\mathbf{E}_j$ as the (i,j)-th block element of $\mathbf{A}$; thus we have

$$\mathbf{I} = \sum_k \mathbf{E}_k\mathbf{E}_k^+ \quad . \tag{4.1}$$

It may be possible to store large matrices in partitioned form, with the natural units to be stored and manipulated being the submatrices constituting the blocks.

What of the other approaches as influenced by the storage problem? The reduction to an equivalent matrix-vector problem advocated by Barakat [4] suffers the same fate as the outer product representation when n is large in that all the matrix elements are not equally accessible. Reference to [4], see Eq. (1), shows that the Roth column decomposition of $\mathbf{AB}$ contains replicas of the matrix $\mathbf{A}$ along the principal diagonal; so that in this version all the matrix elements cannot be held in primary storage. Thus for large n, the parallelism inherent in the general reduction to the Roth column decomposition for matrix-vector multiplications is inhibited. However, there is also a Roth row decomposition of $\mathbf{AB}$, see Eq. (4) of [4], in which the matrix elements of $\mathbf{A}$ are now spread along diagonals. It was hoped, in view of the previous work by Madsen, *et al.* [14] on matrix multiplication by diagonals, that the storage problem could be circumvented. A detailed analysis which we need not reproduce indicates that the row decomposition is no more efficient than the column decomposition as regards the primary storage of matrix elements.

Finally we come to the algorithm of the present paper. The implementation of the algorithm as discussed in Section 3 bears directly upon the storage problem. When the matrices are large enough to violate the equal accessibility condition, we can still maintain a reduced degree of parallelism because the convolution algorithm does not require the rather complicated bookkeeping that the column middle product decomposition necessitates before calculations can be carried out. Even though we cannot simultaneously store *all* the matrix elements of $\mathbf{A}$ and $\mathbf{B}$, the convolution algorithm only requires the rows of $\mathbf{A}$ to be stored on separate optical masks so they can interact with the successive columns (in reverse order) of $\mathbf{B}$ sequentially stand on optical masks to produce the various rows of $\mathbf{C}$. Consequently when both $\mathbf{A}$ and $\mathbf{B}$ are large, we can still maintain a degree of parallelism because we do not require all the matrix elements of $\mathbf{A}$ and $\mathbf{B}$ to be in primary storage simultaneously. All we need in primary storage are the respective row and column of $\mathbf{A}$ and $\mathbf{B}$. Thus, the polynomial convolution algorithm seems to be more immune to the storage problem than do the algorithms in [2,4]. This is because both the outer product and Kronecker product decomposition algorithms are not modular in structure: if the equal accessibility condition is violated there is no way to patch them up to work in the situation where the matrices are very large. It may be possible to employ partitioning as described in [3] or in the present paper; however, the bookkeeping is probably going to be a significant obstacle.

APPENDIX A

The purpose of this appendix is to outline an efficient formalism (due to our colleague D. G. M. Anderson, unpublished) describing the inner product, intermediate product, and outer product representations of matrix multiplication. We further employ this formalism to discuss matrix partitioning, see Eq. (4.1).

To begin we avoid unnecessary complications by assuming that the two matrices, call them $\mathbf{A}$ and $\mathbf{B}$, are square. It is also convenient to use the vector $\mathbf{e}_k$ which is the k-th column of the unit matrix, *i.e.*,

$$\mathbf{I} = \sum_{k=1}^{n} \mathbf{e}_k \mathbf{e}_k^+ \qquad (A.1)$$

and the plus sign denotes the transpose (thus $\mathbf{e}_k^+$ is a row vector). Given the square matrix $\mathbf{A}$, we have

j-th colum of $\mathbf{A} = \mathbf{A}\mathbf{e}_j$

i-th row of $\mathbf{A} = \mathbf{e}_i^+ \mathbf{A}$

(i,j)-th element of $\mathbf{A} = \mathbf{e}_i^+ \mathbf{A}\mathbf{e}_j$.

The usual element representation of the matrix product $\mathbf{C} = \mathbf{AB}$ reads in the above notation

$$\mathbf{e}_i^+ \mathbf{C}\mathbf{e}_j = \sum_k (\mathbf{e}_i^+ \mathbf{A}\mathbf{e}_k)(\mathbf{e}_k^+ \mathbf{B}\mathbf{e}_j \quad . \qquad (A.2)$$

The element representation is the old fashioned way that matrices were multiplied before high level programming languages were invented.

To obtain the *inner product representation*, we begin with the element representation, Eq. (A.2), and delete the parenthesis on the right hand side, thus

$$\begin{aligned}\mathbf{e}_i^+\mathbf{C}\mathbf{e}_j &= \sum_k \mathbf{e}_i^+\mathbf{A}\mathbf{e}_k\mathbf{e}_k^+\mathbf{B}\mathbf{e}_j \\ &= (\mathbf{e}_i^+\mathbf{A})\left[\sum_k \mathbf{e}_k\mathbf{e}_k^+\right](\mathbf{B}\mathbf{e}_j) \\ &= (\mathbf{e}_i^+\mathbf{A})(\mathbf{B}\mathbf{e}_j) \qquad (A.3)\end{aligned}$$

The reason it is termed the inner product representation is that the matrices **A** and **B** are sandwiched between the unit vectors.

At the other extreme, we have the *outer product representation* which we obtain in the following fashion from the element representation, Eq. (A.2):

$$\mathbf{e}_i^+\mathbf{C}\mathbf{e}_j = \sum_k \mathbf{e}_i^+\mathbf{A}\mathbf{e}_k\mathbf{e}_k^+\mathbf{B}\mathbf{e}_j = \mathbf{e}_i^+\left[\sum_k (\mathbf{A}\mathbf{e}_k)(\mathbf{e}_k^+\mathbf{B})\right]\mathbf{e}_j \quad . \qquad (A.4)$$

Consequently

$$\mathbf{C} = \sum_k (\mathbf{A}\mathbf{e}_k)(\mathbf{e}_k^+\mathbf{B}) \quad . \qquad (A.5)$$

The reason it is termed the outer product representation is that the matrices **A** and **B** now reside at the extreme left and right of the summation. This expression can be shown to be equivalent to the expression given in Athale and Collins [2], see their Eq. (2).

We next consider two intermediate representations which we term the *column intermediate product representation* and the *row intermediate product representation*. We return again to Eq. (A.2):

$$\mathbf{e}_i^+\mathbf{C}\mathbf{e}_j = \sum_k \mathbf{e}_i^+\mathbf{A}\mathbf{e}_k\mathbf{e}_k^+\mathbf{B}\mathbf{e}_j = \mathbf{e}_i^+ \sum_k (\mathbf{A}\mathbf{e}_k)[\mathbf{e}_k^+(\mathbf{B}\mathbf{e}_j)] \qquad (A.6)$$

or

$$\mathbf{C}\mathbf{e}_j = \sum_k (\mathbf{A}\mathbf{e}_k)[\mathbf{e}_k^+(\mathbf{B}\mathbf{e}_j)] \quad . \tag{A.7}$$

This is the column intermediate product. The corresponding row intermediate product is

$$\mathbf{e}_i^+ \mathbf{C}\mathbf{e}_j = \sum_k [(\mathbf{e}_i^+ \mathbf{A})\mathbf{e}_k](\mathbf{e}_k^+ \mathbf{B})\mathbf{e}_j \tag{A.8}$$

or

$$\mathbf{e}_i^+ \mathbf{C} = \sum_k [(\mathbf{e}_i^+ \mathbf{A})\mathbf{e}_k](\mathbf{e}_k^+ \mathbf{B}) \quad . \tag{A.9}$$

It is a straightforward exercise to extend the above formalism to accommodate rectangular matrices, we omit the details.

ACKNOWLEDGEMENTS

Barakat was supported in part by AFOSR under Contract F49620-85-C-001 with RGB Associates, Inc. In addition, Barakat was supported (through RGB Associates, Inc.) by the Innovative Science and Technology Office for the Strategic Defense Initiative Organization and was administered through the Office of Naval Research under Contracts N00014-85-K-0479 and N00014-86-K-0591.

REFERENCES

1. D. Casasent and C. Neumann, "Iterative optical vector-matrix processors," in: *Optical Information Processing for Aerospace Applications*, NASA Conference Publication 2207 (NTIS, Springfield, VA, 1981), p. 105.

2. R. A. Athale and W. C. Collins, "Optical matrix-matrix multiplier based on outer product decomposition," *Appl. Opt.* **21**, 2089 (1982).

3. H. Caulfield, C. Verber, and R. Stermer, "Efficient matrix partitioning for optical computing," *Opt. Commun.* **51**, 213 (1984).

4. R. Barakat, "Optical matrix-matrix multiplier based on Kronecker product decomposition," *Appl. Opt.* **26**, 191 (1987).

5. A. R. Dias, "Incoherent optical matrix-matrix multiplier," in: *Optical Information Processing for Aerospace Applications*, NASA Conference Publication 2207 (NTIS, Springfield, VA, 1981), p. 71.

6. W. K. Cheng and H. Caulfield, "Fully-parallel relaxation algebraic operations for optical computers," *Opt. Commun.* **43**, 251 (1982).

7. R. Bocker, H. Caulfield, and K. Bromley, "Rapid unbiased bipolar incoherent calculator cube," *Appl. Opt.* **22**, 804 (1983).

8. R. Bocker, "Advanced RUBIC cube processor," *Appl. Opt.* **22**, 2401 (1983).

9. R. Bocker, K. Bromley, and S. Clayton, "A digital optical architecture for performing matrix algebra," *SPIE* **431**, 194 (1983).

10. H. Nakano and K. Hotate, "Optical system for real-time processing of multiple matrix product," *Electron. Lett.* **21**, 436 (1985).

11. S. Cartwright and S. Gustafson, "Convolver-based optical systolic processing architectures," *Opt. Eng.* **24**, 59 (1985).

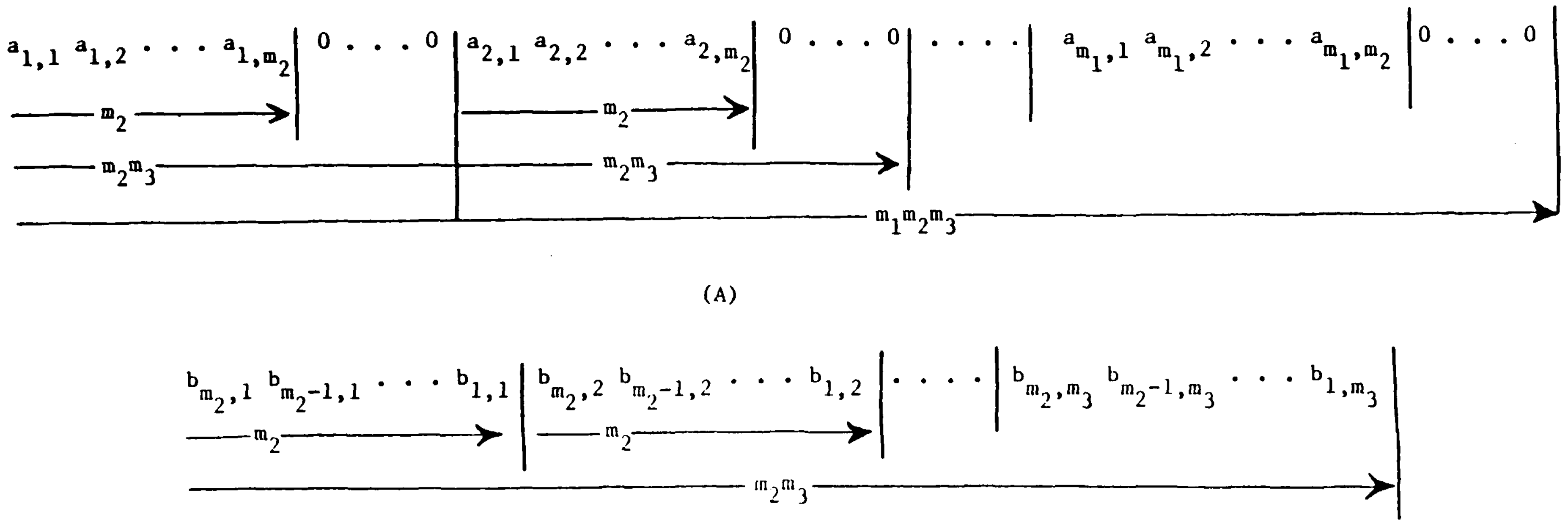
12. J. Goodman, "Architectural development of optical data processing systems," *Kinam* 5C, 9 (1983).

13. R. W. Hockney and C. R. Jesshope, *Parallel Computers: Architecture, Programming and Algorithms* (Adam Hilger, Bristol, 1981), pp. 276–279.

14. N. Madsen, G. Rodrigue, and H. Karush, "Matrix multiplication by diagonals on a vector/parallel processor," *Inf. Process. Lett.* 5, 41 (1976).

Table 1. Listing of the p, q and r coefficients for the case where **A** is 2 × 2, **B** is 2 × 3 and **C** is 2 × 3.

	p_s	q_t	r_m
0	a_{11}	b_{21}	
1	a_{12}	b_{11}	c_{11}
2	0	b_{22}	
3	0	b_{12}	c_{12}
4	0	b_{23}	
5	0	b_{13}	c_{13}
6	a_{21}		
7	a_{22}		c_{21}
8	0		
9	0		c_{22}
10	0		
11	0		c_{23}
12	0		


$a_{1,1}$ $a_{1,2}$ . . . a_{1,m_2}
0 . . . 0
$a_{2,1}$ $a_{2,2}$ . . . a_{2,m_2}
0 . . . 0
. . . .
$a_{m_1,1}$ $a_{m_1,2}$ . . . a_{m_1,m_2}
0 . . . 0
m_2
m_2
m_2m_3
m_2m_3
$m_1m_2m_3$
$b_{m_2,1}$ $b_{m_2-1,1}$ . . . $b_{1,1}$
$b_{m_2,2}$ $b_{m_2-1,2}$ . . . $b_{1,2}$
. . . .
b_{m_2,m_3} b_{m_2-1,m_3} . . . b_{1,m_3}
m_2
m_2
m_2m_3

(A)

(B)

Fig. 1. Layout of the **p** vector, see (A), and the **q** vector, see (B).